UNIVERSIDADE FEDERAL DE SANTA CATARINA
PROGRAMA DE PÓS-GRADUAÇÃO EM CIÊNCIA DA COMPUTAÇÃO

Fábio Alexandre Taffe

ANÁLISE DE REQUISITOS E FERRAMENTAS PARA IMPLEMENTAÇÃO DE SITES DE COMÉRCIO ELETRÔNICO MÓVEL

Dissertação submetida à Universidade Federal de Santa Catarina como parte dos requisitos para a obtenção do grau de Mestre em Ciência da Computação

Orientador:

Mauro Roisenberg

Florianópolis, Março de 2002

“Para alcançar grandes conquistas,
não devemos apenas agir – temos também
que sonhar. Não basta apenas planejar –
temos que acreditar.”
Anatole France

OFERECIMENTO

À minha Mãe, um exemplo de MULHER.

AGRADECIMENTOS

À Deus,
por estar comigo em todos os momentos de minha vida me iluminado, e por mais que em alguns momentos me desviasse do seu caminho, jamais me abandou.

À toda minha Família,
o maior presente que Deus me deu, que é o principal motivo de continuar sempre lutando, meu eterno agradecimento.

Ao meu orientador Prof. Dr. Mauro Roisenberg,
pela força e confiança depositada, meu muito obrigado.

Aos verdadeiros amigos,
que me acompanharam nesta caminhada e contribuíram para que pudesse chegar ao fim de mais esta batalha.

E finalmente, a todos que diretamente ou indiretamente estiveram ao meu lado, contribuindo com experiências pessoais, motivação e otimismo, meu agradecimento.

RESUMO

Com o advento do Comércio Eletrônico mudaram alguns conceitos na forma de se fazer negócio, porém, essas facilidades eram oferecidas praticamente a usuários fixos. Informações móveis e usuários fixos. Através de propostas, sugestões, motivações e mudanças aplicadas ao comércio eletrônico tradicional, originou uma nova modalidade de comércio, o Comércio Eletrônico Móvel ou *Mobile Commerce*, devido à rápida disseminação da tecnologia sem fio e o grande número de empresas participantes, várias ferramentas foram desenvolvidas sem a preocupação de se criar um padrão unificado para utilização. Partindo desse pressuposto, o presente trabalho visa desenvolver uma análise das ferramentas existentes no mercado, dando uma ampla visão dos requisitos necessários para a implementação de sites para o comércio eletrônico móvel. Abordando aspectos teóricos e práticos, iniciando com um embasamento da tecnologia sem fio e suas aplicações, direcionando ao comércio eletrônico. Em seguida, apresenta a arquitetura *wireless*, requisitos de configuração de servidores e o problema enfrentado na questão segurança e os cuidados a serem tomados. Para um melhor aproveitamento do estudo envolvido no presente trabalho, desenvolveu-se um protótipo de comércio eletrônico móvel, utilizando a linguagem de marcação WML e as ferramentas analisadas, tomando por base custo-benefício, qualidade e funcionalidade dos serviços.

Palavras-chave: wireless, wap, ferramentas, análise de requisitos, sem fio, dispositivos móveis e comércio eletrônico.

ABSTRACT

With the advent of the Electronic Commerce they had changed some concepts in the form of if to make business, however, these easinesses were practically offered the fixed users. Mobile information and using fixtures. Through proposals, suggestions, motivations and changes applied to the traditional electronic commerce, originated a new modality of commerce, the Electronic Commerce Mobile or Mobile Commerce, due to fast use of the technology wireless and the great number of participant companies, some tools had been developed without the concern of if creating a standard unified for use. Leaving of this estimated, the present work aims at to develop an analysis of the existing tools in the market, giving an ample vision of the necessary requirements for the implementation of sites for the mobile electronic commerce. Approaching theoretical and practical aspects, initiating with a basement of the technology wireless and its applications, directing to the electronic commerce. After that, it presents the architecture wireless, requirements of pattern of servers and the problem faced in the question security and the cares to be taken. For a better exploitation of the involved study in the present work, developed an archetype of mobile electronic commerce, using the analyzed language of marking WML and tools, taking for base cost-benefit, quality and functionality of the jobs.

Keywords: wireless, wap, tools, analysis of requirements, mobile devices and electronic commerce.

SUMÁRIO

LISTA DE ILUSTRAÇÕES

LISTA DE TABELAS

Capítulo 1

INTRODUÇÃO

1.1 Motivação

Considerando as diversas turbulências ambientais, caracterizadas pelas constantes mutações dos diversos tipos de tecnologias, bem como seu ciclo de vida cada vez mais reduzido em função da acirrada concorrência de empresas com expressão internacional na área tecnológica, aliado a um consumidor cada vez mais refinado em seus desejos e necessidades, e os inerentes altos investimentos em pesquisa e desenvolvimento e incertezas associadas a novas concepções ligadas a tecnologia, pode-se afirmar que estar desenvolvendo e lançando ao mercado produtos com alto valor agregado geram muitas incertezas e riscos para as empresas. Apesar da impossibilidade de previsões mais acertadas a cerca da probabilidade de sucesso, além do que os usuários de tais serviços ou tecnologias ainda tem arraigado culturalmente a utilização de meios tradicionais de transações, surgiu o comércio eletrônico que aos poucos foi ganhando espaço, aprimorando serviços e ferramentas, dando uma maior credibilidade a esse serviço.

Facilidade de atingir seu público-alvo, quebrando barreiras geográficas, disponibilizando ao seu cliente funcionamento de seu comércio 24 horas por dia, 7 dias na semana, entre outras vantagens. Porém, continuando preso a um local “fixo” (na maioria dos casos em um *desktop*). Através de propostas de mudanças, melhoras e motivações, abriu espaço a um novo tipo de comércio, uma nova forma de comércio eletrônico que daria um fim a toda essa imobilidade. Surgindo então, o comércio eletrônico móvel, ou *mobile commerce* que ainda não conquistou um espaço suficientemente grande para gerar credibilidade, devido ao custo de troca de uma tecnologia para outra, que ainda é grande em função da não visualização de um custo-benefício maior, principalmente por parte das grandes corporações.

Neste sentido, existe a questão de necessidade de adequação às tendências de mercado, onde apesar das incertezas geradas pelos diversos tipos de tecnologias e suas

aplicações, aliadas ao temor gerado pelo novo, tanto em empresas quanto em usuários tradicionais, tem-se um panorama bastante complexo para as organizações que disputam fatias de mercado em um segmento em visível ascensão. Impondo o imperativo de busca por mecanismos mais refinados de prospecção, refinamento de informações captadas no mercado *wireless* e traduzindo em benefícios para os usuários finais.

A tecnologia considerada no presente trabalho, o WAP – Wireless Application Protocol (Protocolo de Aplicações Sem Fio), utilizada como padrão para comunicação sem fio, gerenciado pelo WAP FORUM, apesar dos problemas apresentados, principalmente devida limitação de equipamentos e incompatibilidade vem sendo um elo de ligação muito importante que pode levar ao sucesso ou não da tecnologia sem fio.

1.1.1 Objetivo Geral

"Análise de Requisitos e Ferramentas para implementação de Sites de Comércio Eletrônico Móvel."

1.1.2 Objetivo Específico

O presente trabalho tem por objetivo analisar os requisitos e as ferramentas para implementação de sites de comércio eletrônico móvel, sendo para isso desenvolvido um protótipo para uma análise mais completa das reais necessidades para seu funcionamento. Fornecendo aos desenvolvedores e analistas de negócios, subsídios concretos do potencial da comunicação móvel. Partindo da motivação apresentada e do objetivo geral podemos dividi-los da seguinte forma para uma melhor visualização:

"Expor sobre o crescimento do mercado da comunicação sem fio"

"Demonstrar os requisitos necessários para implementação de aplicativos móveis"

"Verificar os elementos utilizados para uma programação limitada aos recursos existentes no mercado móvel"

“Analisar ferramentas para implementação de sites móveis”

“Desenvolver um protótipo aplicando conhecimentos adquiridos”

1.2 Organização do Trabalho

O presente trabalho está distribuído em sete capítulos, conforme segue:

Capítulo 1 - Introdução, onde são apresentadas as motivações que levaram ao desenvolvimento do trabalho, bem como os objetivos gerais e específicos a serem atingidos e a distribuição dos capítulos.

Capítulo 2 – Comércio Eletrônico, apresenta conceitos de comércio eletrônico e fatores que impulsionaram o e-commerce para ser adaptado ao mercado *Wireless*. Desmistificando as tecnologias existentes no mercado mundial em redes sem fio e suas áreas de aplicação, advertindo sobre os cuidados necessários para implementação de *sites* de comércio eletrônico móvel.

Capítulo 3 – Internet Móvel, aborda amplamente o WAP (Protocolo de Aplicações Sem Fio), como surgiu, objetivo principal e sua arquitetura. Descreve as configurações que devem ser feitas nos principais servidores existentes, para que os sites sejam visualizados corretamente nos dispositivos móveis.

Capítulo 4 - WML, onde é abordada a linguagem de marcação utilizada em dispositivos sem fio, comandos utilizados, cuidados com formatação de textos. Descreve a linguagem WMLScript, responsável pelo aumento velocidade dos processos, devido às bibliotecas residentes no *browser* do dispositivo do usuário. Inserção de imagens estáticas e o método utilizado para geração imagens animadas, funcionamento de temporizadores e *cache*.

Capítulo 5 – Ferramentas para Implementação do M-Commerce, são apresentadas as principais ferramentas existentes no mercado para implementação do comércio eletrônico móvel. Entre essas ferramentas são citadas: os editores para linguagem WML, editores de imagens, micronavegadores e emuladores.

Capítulo 6 - Protótipo, utilizando a base teórica apresentada no decorrer do trabalho, desenvolveu-se um modelo de comércio eletrônico móvel, criando uma “pizzaria virtual móvel”. Onde as ferramentas escolhidas atenderiam alguns

requisitos como: custo-benefício aliado à qualidade e garantia de funcionamento dos serviços prestados.

Capítulo 7 - Conclusão, apresentação dos resultados colhidos a partir das análises de requisitos e ferramentas utilizadas para implementação de sites de comércio eletrônico, onde são apresentadas a vantagens e desvantagens para implementação desse novo segmento de comércio. Demonstrando suas falhas e propondo implementações para trabalhos futuros.

Capítulo 2

COMÉRCIO ELETRÔNICO

Com a popularização da Internet, não somente nos meios acadêmicos e usos domésticos, como forma de troca de informações. Mas também como uma ferramenta de suma importância para o desenvolvimento, transformando completamente a economia e a sociedade em todo mundo. A cada dia que passa mais empresas buscam na internet um meio de tirar vantagens nas mais diversas áreas de atuação para ampliar sua gama de clientes, partindo desse pressuposto, surge o Comércio Eletrônico.

Segundo ALTERNTIN (2000), Comércio Eletrônico é definido da seguinte forma:

> **"***Comércio eletrônico é a realização de toda a cadeia de valor dos processos de negócio num ambiente eletrônico, por meio da aplicação intensa das tecnologias de comunicação e de informação, atendendo aos objetivos de negócio. Os processos podem ser realizados de forma completa ou parcial, incluindo as transações negócio-a-negócio, negócio-a-consumidor e intra-organizacional, numa infra-estrutura predominantemente pública de fácil e livre acesso e baixo custo".*

Direcionando este conceito para Ciências da Computação, MEIRA (2000) define Comércio Eletrônico, sendo um conjunto de técnicas e tecnologias computacionais utilizadas para facilitar e executar transações comerciais de bens e serviços físicos ou virtuais. Onde as técnicas e tecnologias empregadas para estas transações, dependem da complexidade e dos custos dos sub-processos.

2.1 Comércio Eletrônico na Internet

Para MEIRA (2000), o sucesso do Comércio Eletrônico deve-se a três aspectos fundamentais. Primeiro, a facilidade da disseminação de informações na internet, onde os produtos são expostos de maneira abrangente ao público, em quantidade superior ao comércio tradicional (físico) e interação com seu público-alvo (através de pesquisas no próprio *site*). Segundo, o baixo valor das transações eletrônicas, que reduz consideravelmente custos como: utilização de telefone, fax, material de apoio, e devido essas reduções podem ser repassados aos preços finais dos produtos e serviços, tornando-se mais competitivo se comparado ao comércio eletrônico tradicional. E por último, competição pelos clientes, onde empresas que prestam serviço de comércio eletrônico devem ficar atentos a seus futuros clientes, pois, efetuando uma simples pesquisa de preço, podem decidir sobre a compra ou não de determinado produto ou até mesmo a compra em seu concorrente.

Segundo ALBERTIN (1999), são muitas as vantagens de possuir um comércio eletrônico para sua empresa como: independência de barreiras físicas, ampliação de seu mercado consumidor, disponibilização de informações completas e atualizadas de produtos e serviços, facilidade de atingir seu público-alvo, produtos e serviços disponíveis aos consumidores 24 horas por dia, prestação de serviços dos produtos vendidos pela empresa física e venda direta de produtos.

2.2 Segurança na Web

Para garantir a confiabilidade aos usuários nas transações efetuadas na internet, através do comércio eletrônico, um ponto fundamental deve ser levado em consideração, a segurança.

O protocolo utilizado na internet que desempenha este papel é o SSL (Secure Sockets Layer), que garante uma conexão segura entre o cliente e o servidor, sem a intervenção de qualquer intermediário. Ao estabelecer uma sessão é evitada a negociação dos parâmetros de segurança a cada conexão, sendo também efetuada uma autenticação pelas partes envolvidas e a negociação do processo de criptografia, conforme mostra figura 2.1 (DENEGA, 2000, TERRY et al, 1997).

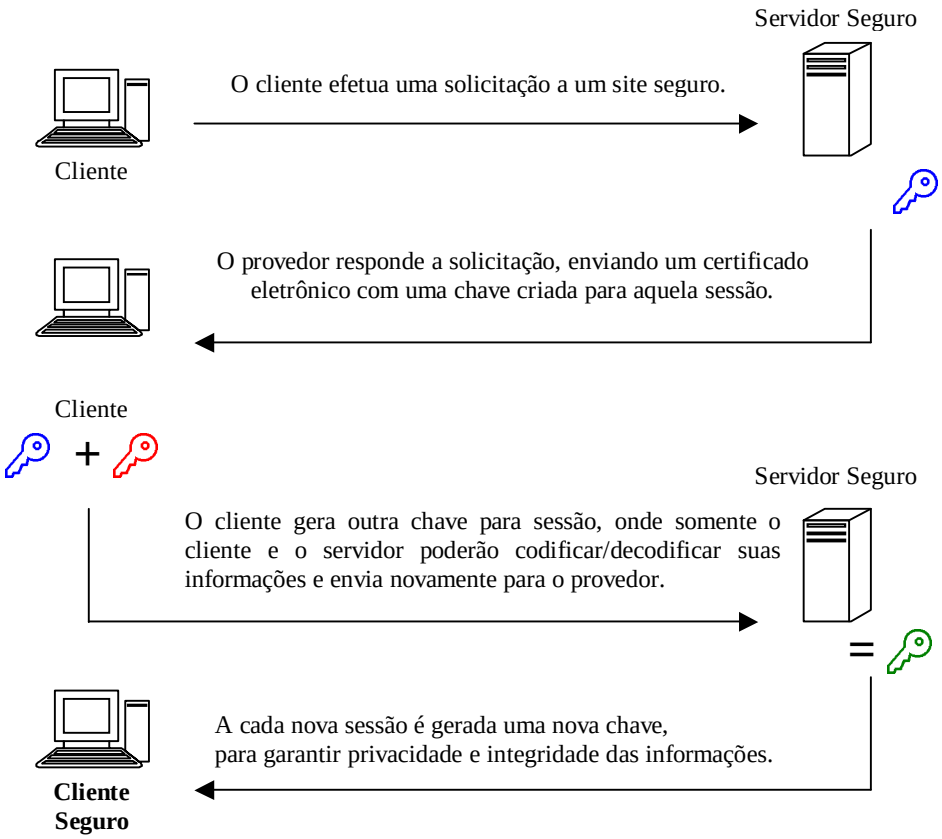

Figura 2.1– Segurança na Web

2.3 Arquitetura da Rede Sem Fio

A arquitetura de uma rede sem fio é composta basicamente pelos seguintes elementos (DORNAN, 2001, HARTE, 2001): BSC (Base Station Controller), MSC (Mobile Switching Center), EIR (Equipment Identity Register), HLR (Home Location Register), VLR (Visitor Location Register), conforme mostra a figura 2.2.

Sendo as torres de rádio os locais de transmissão da telefonia celular (móvel). O BSC (Controlador da Estação Base), tem como principais funções: preparar a chamada de voz ou de dados de um dispositivo móvel e ser responsável pela mudança de torres ainda com o aparelho *on-line*, sem comprometimento do serviço. Já o MSC (Centro de Comutação Móvel), tem a responsabilidade de rastrear usuários, efetuar envio de chamadas e bilhetagem. Na sua rede sem fio pode conter vários MSC, onde cada

usuário é registrado em sua MSC local, sendo esta responsável por um determinado número de células. AuC (Centro de Autenticação) está presente apenas nas redes digitas, sendo responsável pela autenticação e validação dos serviços dos dispositivos que estão utilizando a rede. Impedindo também a clonagem de aparelhos através de mecanismos de segurança. A base de dados é dividida da seguinte forma:

- EIR (Registro de Identidade de Equipamentos): responsável pela verificação e armazenamento do ESN (Eletronic Serial Number – número de série eletrônico) e pelo MIN (Moblile Identity Number – número identificador móvel), utilizado para bloquear a utilização do aparelho quando o mesmo for roubado. Efetua a bilhetagem das ligações;
- HLR (Registro de Localização na área original): possui a localização dos aparelhos registrados no MSC, onde os dispositivos informam (se registram) numa rede sem fio para identificar sua localização, e caso sair de sua área o HLR armazena o número do MSC para onde foi deslocado;
- VLR (Registro de Localização de Visitante): registra a localização de um aparelho visitante que se encontram na área de cobertura de um MSC. Todas as ligações são roteadas pelo HLR local do usuário, tornando o custo dessa operação alto, por exemplo, num *roaming* internacional.

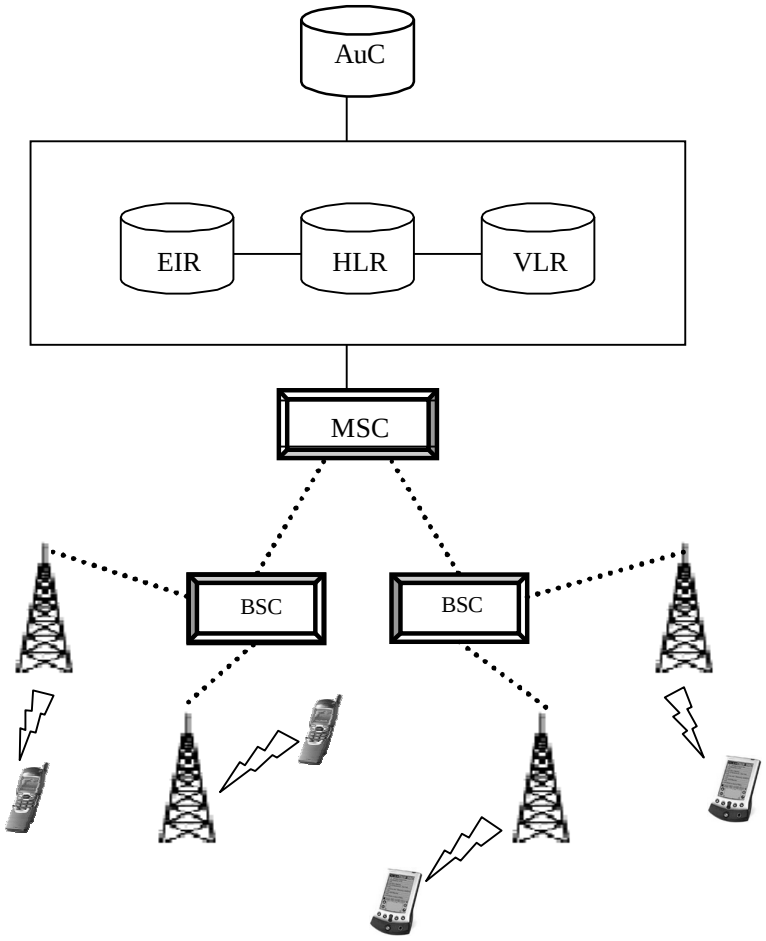

Figura 2.2 – Arquitetura Básica de Rede Sem Fio

A transmissão de dados numa rede sem fio é efetuada de formas: transmissão de dados comutada por pacote (*packet-switched*) ou por circuito (*circuit-switched*). Na transmissão de dados por pacote, os dados são separados em pequenos pacotes e posteriormente enviados com o endereço de destino fixado em cada unidade. E logo após chegar ao seu destino, as informações são remontadas. Neste tipo de transmissão é cobrada pelo número de pacotes enviados. Já na transmissão de dados por circuito, é efetuada uma conexão dedicada entre as duas partes envolvidas no processo. Utilizada normalmente para transmissões de grande quantidade de dados, sendo a cobrança na transmissão por circuito feita pelo tempo de conexão (HEIJDEN, 2000).

2.4 Geração de Telefones Celulares

Segundo DORNAN (2001), as gerações de telefones celulares dividem-se da seguinte forma:

- 1G: transmissão analógica, tecnologias: AMPS (Advanced Mobile Phone System) nos Estados Unidos, TACS (Total Access Communication System) na Europa, NMT (Nordic Mobile Telephone) no Japão. Possuem baixa qualidade nas chamadas, problemas com a segurança e fragilidade no sistema, são marcas da primeira geração de aparelhos móveis;
- 2G: transmissão digital (em sua maioria), converte som em código digital dando maior qualidade às chamadas, maior segurança (utiliza-se de sistemas de criptografia), possuem recursos de identificação de chamada e correio de voz. Tecnologias utilizadas: TDMA (Time Division Multiple Access) e CDMA (Code Division Multiple Access) nos Estados Unidos, GSM (Global System for Mobile Communications) na Europa, PDC (Personal Digital Cellular) no Japão. Transmissão de dados de aproximadamente 10 Kbps;
- 2,5G: aumento da capacidade de transferência de dados, utilizando tecnologia como: EDGE (Enhanced Data Rates for Global Evolution) até 384 Kbps, GPRS (General Packet Radio Service) até 144 Kbps, HSCSD (High Speed Circuit-Switched Data) até 38,4 Kbps;
- 3G: ampliação da gama de serviços móveis disponíveis aos usuários, devido ao aumento da taxa de transferência de dados até 2 Mbps. Utilização de dispositivos móveis avançados com telas coloridas, transmissão de som e vídeo. Conexão dos dispositivos móveis com a estação-base WAN (Wide Area Network) e LAN (Local Area Network);
- 4G: sendo atualmente testada em laboratórios, permitirá taxas de transferências de até 150 Mbps, baixa taxa de erros de bits e qualidade nos serviços. Promissor serviço de telepresença, semelhante à realidade virtual, onde é passada a sensação de um objeto real.

O Brasil encontra-se num momento de transição entre a segunda geração (2G) e a geração 2,5G. Sendo lançada pela Telesp Celular com tecnologia CDMA 1X, com

velocidade de 144 Kbps. O Japão encontra-se atualmente na terceira geração (3G) de aparelhos celulares, com um número superior a 27 milhões de usuários. Ao contrário do restante do mundo que aderiu ao protocolo WAP (Wireless Application Protocol) como padrão mundial, criou o iMode (da NTT DoCoMo), que utiliza tecnologia W-CDMA (Wideband Code Division Multiple Access). Com velocidade atual de 384 Kbps, possui recursos avançados de som, vídeo e conexão permanente a Web (INFO, 2001).

2.5 Mercado Wireless

Com a rápida disseminação dos dispositivos sem fio, a falta de uma estruturação adequada, para que fosse utilizada uma única tecnologia para todos os dispositivos, dependência do modelo de negócio adotado, tendências políticas de cada país, onde o fator financeiro deve ser levado em consideração, como por exemplo, a geração 2,5G que adaptou a estrutura existente e aprimorou tecnologias, ganhando com isso velocidade e atendendo a necessidade atual antes de partir para 3G, onde os custos são elevados (HARTE, 2001). Todos esses fatores aliados ao próprio avanço tecnológico, deram origem a diferentes tecnologias e mercados voltados para comunicação *wireless* em todo mundo, que são explicados a seguir (DORNAN, 2001, RISCHPATER, 2001) :

- AMPS (Advanced Mobile Phone System – Sistema Avançado de Telefonia): padrão de telefonia móvel analógico utilizado na América do Norte e Sul e parte da Ásia;
- CDMA (Code Division Multiple Access – Acesso Múltiplo por Divisão de Código): método de compartilhamento de freqüência, onde é atribuído aos sinais de transmissão e recepção um código de espectro largo (wide spectrum);
- CDMA2000 : grupo de tecnologias que atualizam as redes cdmaOne para transferências de dados de no mínimo 2 Mbps;
- EDGE (Enhanced Data Rates for Global Evolution – Taxa de Transmissão Aprimorado para Evolução Global): melhora a velocidade de dados para redes GSM, é compatível com sistemas TDMA como D-AMPS e o PDC. Desenvolvido para transferência de grande quantidade de dados;

- GPRS (General Packet Radio Service – Serviço Padrão para Transmissão de Pacotes Via Rádio): atualização para redes GSM que oferece a cada usuário até oito canais de 14,4 Kbps, utilizando comutação de pacotes. Juntamente com o EDGE, são opções alternativas antes da implementação da 3G, que possui estrutura relativamente cara se comparada a 2,5G;
- GSM (Global System for Mobile Communications – Sistema Global para Comunicação Móvel): padrão TDMA de banda larga, utilizando canais com largura de 200Khz com oito usuários para cada canal. Originado na Europa, porém, sendo utilizando em todo mundo;
- HSCSD (High Speed Circuit-Switched Data – Dados Comutados por Circuito de Alta Velocidade): software de atualização para redes GSM que oferece a cada usuário até quatro circuitos de 14,4 Kbps. São assimétricos, permitindo maior velocidade de *downloads* do que para *uploads*;
- IMT2000 (International Mobile Telecommunications – Telecomunicações Móveis Internacionais 2000): projeto da ITU (International Telecommunications Union), com o objetivo de criar padrões para o acesso global sem fio de terceira geração (3G). Incluindo 3 sistemas: UMTS/W-CDMA, EDGE/UWC136 e cdma2000, oferecendo dados em comutação de pacotes, com velocidade de até 2 Mbps quando fixa e 384 Kbps quando móvel;
- NMT (Nordic Mobile Telephone – Padrão de Telefone Móvel Nordico): padrão utilizado pelo celular analógico no Japão;
- PDC (Personal Digital Cellular - Celular Digital Pessoal): padrão utilizado pelo celular digital japonês. Utilizando o iMode, um dos melhores sistemas de internet móvel desenvolvido atualmente, criado pela operadora NTT DoCoMo;
- TACS (Total Access Communication System – Sistema de Comunicação de Acesso Total): sistema de comunicação analógico utilizado em alguns locais na Europa e Ásia, que será substituído pelo GSM;
- TDMA (Time Division Multiple Access – Acesso Múltiplo por Divisão de Tempo): utiliza compartilhamento de mesma freqüência entre vários assinantes, sendo limitado os sinais de transmissão e recepção em intervalos de tempo;

- UMTS (Universal Mobile Telecommunications System – Sistema Universal de Telecomunicações Móveis): padrão europeu de terceira geração, que utiliza W-CDMA;
- WATM (Wireless Asynchronous Transfer Mode – Modo de Transmissão Assíncrono Sem Fio): tecnologia de quarta geração, ainda em laboratório, podendo alcançar velocidade entre 10 Mbps a 150 Mbps, possui baixa taxa de erro de bits e serviços de alta qualidade.
- W-CDMA (Wideband CDMA – CDMA de Banda Larga): tecnologia utilizada na maioria dos sistemas de terceira geração. Suporte para serviços de sons e imagens animadas, vídeo em tempo real e videoconferência.

2.6 Aplicações Comerciais da Internet Sem Fio - WIEA

Aplicações comerciais geralmente são relacionadas a serviços, podendo ser: horizontais ou verticais. As Aplicações Comerciais Horizontais, são quando aplicações suprem somente às necessidades de um grupo. Já as Aplicações Comerciais Verticais, atendem às necessidades específicas (por exemplo, uma empresa gerenciando a área de controle de estoque). Unindo essas duas soluções para serem fornecidas e utilizadas em dispositivos sem fio, surgem os aplicativos comerciais para internet sem fio (SHARMA 2001). A tabela 2.1 exemplifica algumas aplicações comerciais para internet sem fio:

Informações	**M-Commerce**
Notícias	Compra de Produtos
Meteorologia	Bancos
Restaurante	Serviço ao Cliente
Horário de Vôos	Planejamento de Recursos da Empresa
Taxas de Câmbio	Serviços de Localização
Médica	**Entretenimento**
Monitoração de Cuidados Médicos	Programação de Cinema/Teatros
Receitas Médicas	Jogos Interativos Multiusuários
Diagnósticos remotos	Bate-Papo
Serviços de Emergência (localização)	Música
Intranets	**Telemetria**
Automação da Equipe de Vendas	Alarmes contra roubo e incêndio
Diagnóstico e análise remota por rede	Máquinas de venda automática
Videoconferência	Estatísticas de tráfego de veículos
Acesso a e-mails e agendas	Medições de: água, eletricidade, gás

Tabela 2.1 – Aplicações Comerciais Sem Fio

2.7 Projetos – Requisitos e Implementação

No desenvolvimento de aplicativos sem fio, FORTA et al (2000) alerta sobre os cuidados especiais que os projetistas e desenvolvedores devem tomar:

- Largura de banda limitada: devido à baixa largura de banda os arquivos devem possuir tamanhos limitados (no máximo 2 Kb);
- Tamanho de tela reduzido: em média os dispositivos possuem quatro linhas de textos em sua tela e aceitam aproximadamente doze caracteres por linha;
- Diferença de dispositivos: por possuir uma grande variedade de dispositivos móveis, um *site* desenvolvido num aparelho, pode não alcançar o resultado esperado em outro. Por isso, devem ser testados em vários modelos de dispositivos e emuladores.

Para SHARMA (2001), após a identificação do projeto em alto nível e da definição estratégica, deve-se passar para a fase de requisitos (onde é feita à análise e definição dos requisitos necessários para implementação do projeto), somente então, iniciar a implementação do projeto. Na tabela 2.2 define uma lista para implementação na fase de requisitos:

1. Desenvolvimento do protótipo inicial
2. Criação dos planos comerciais
3. Análise da situação do marketing
4. Avaliação da concorrência
5. Pesquisa de mercado
6. Análise de usuário
7. Definição de estratégia criativa
8. Execução da avaliação técnica
9. Definição da estratégia técnica
10. Manutenção de grupo de foco
11. Previsão da aceitação do usuário
12. Requisitos comerciais detalhados
13. Definição de requisitos funcionais
14. Atualização do Plano de Negócios
15. Definição de recursos de alto nível
16. Desenvolvimento da arquitetura técnica conceitual
17. Definição da Interface do usuário
18. Criação do protótipo
19. Definição dos requisitos de software técnicos
20. Criação de casos de teste
21. Definição dos requisitos de desempenho
22. Projeto dos processos comerciais

Tabela 2.2 – Fase de Requisitos

A seguir a tabela 2.3 exibe uma lista para implementação do projeto:

1. Iniciar projeto em alto nível
2. Criar o mapa do *site*
3. Conceituar o projeto visual
4. Criar o mapa de navegação
5. Desenvolver a matriz do conteúdo
6. Definir as ferramentas de teste
7. Criar o plano de fornecimento de conteúdo
8. Engajar-se no desenvolvimento
9. Engajar-se no teste regular de unidade
10. Executar as revisões intermediárias e a análise de utilidade
11. Congelar o código
12. Executar o teste de stress
13. Corrigir os erros
14. Fazer um lançamento limitado
15. Fazer o lançamento em escala completa

Tabela 2.3 – Fase de Implementação

Ainda segundo SHARMA (2001), alguns fatores devem ser levados em consideração para que os projetos para dispositivos sem fio tenham sucesso como:

- Estratégia e objetivos bem definidos;
- Agregar conteúdo ao *site*;
- Garantir interoperabilidade;
- Estar apto a mudanças a novas tecnologias;
- Criação de protótipos e respectivos testes;
- Personalização dos serviços;
- Manter parcerias estratégicas:
- Garantir segurança;
- Qualidade;
- Simplicidade.

2.8 Crescimento da Tecnologia Móvel

O crescimento da tecnologia móvel tem crescido de forma exponencial nos últimos anos (conforme mostra figura 2.3 – SHARMA (2001)), e prova disso são os altos investimentos feitos pelas operadoras, fabricantes de software, desenvolvedores de

conteúdo, portais e fornecedores de infra-estrutura, revolucionando novamente o Comércio.

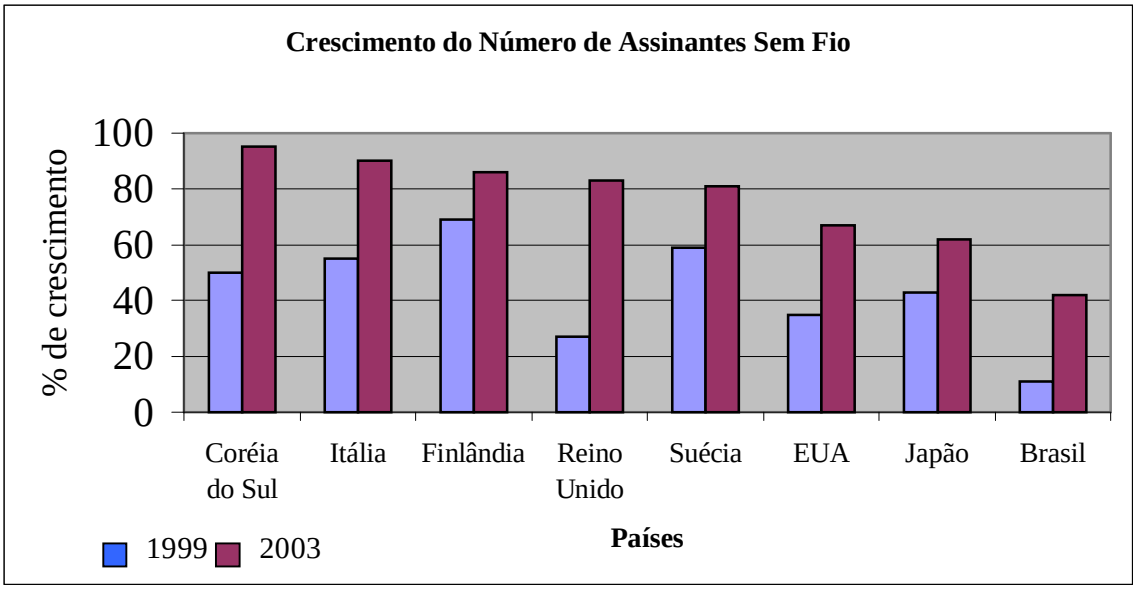

Figura 2.3 – Arquitetura Básica de Rede Sem Fio

O Comércio que vem acompanhando esse progresso tecnológico, partindo do comércio tradicional, e através de melhorias, propostas e motivações passou a ser eletrônico (Comércio Eletrônico). Hoje somando todas as vantagens apresentadas em relação ao comércio eletrônico, aliado a tecnologia da internet sem fio, e através de melhorias e necessidades do mercado atual, surge uma nova modalidade de comércio eletrônico, o M-Commerce (Mobile Commerce).

Capítulo 3

INTERNET MÓVEL

3.1 WAP

Wireless Application Protocol – Protocolo de Aplicações Sem Fio, utilizado como padrão para comunicação sem fio, gerenciado pelo WAP Fórum. O WAP está para os dispositivos móveis, assim como, o HTTP está para os navegadores da Web. É um protocolo para transporte de dados, sendo criado com base nos padrões estabelecidos como: IP, URL e XML (FACUNTE, 2000, AREHART et al, 2000).

SETUBAL (2000) salienta, que o WAP atende exclusivamente a comunicação sem fio, levando em consideração fatores como:

- Dispositivos com pouca memória (RAM e ROM);
- Processamento limitado;
- Baterias de potência e vida útil limitada;
- Telas de pequeno porte;
- Baixa velocidade e largura de banda;
- Interface limitada com o usuário;
- Maior latência na transmissão de dados;
- Instabilidade nas conexões.

"WAP é uma especificação global e aberta que dá o poder aos usuários móveis para acessar de forma fácil e interativa com informações e serviços instantaneamente".

3.2 WAP Fórum

O WAP Fórum, foi criado em 1997 pela Ericsson, Motorola, Nokia e a Unwired Planet (atualmente Phone.com), com o objetivo de assegurar a interoperabilidade do

produto e o crescimento do mercado dos dispositivos sem fio. Estabelecendo um padrão único a ser seguido pelas empresas associadas. Atualmente, fazem parte do WAP Fórum as principais empresas de: hardware, software, dispositivos móveis, provedores e operadoras de telecomunicação. Sendo principal fonte de documentação e especificações relacionadas a WAP (WAPFORUM, 1999).

3.3 A Arquitetura

A arquitetura WAP tem origem do modelo OSI (Open Systems Interconnection), sendo introduzida pela ISO (International Standards Organization).

Segundo TANENBAUM (1996), a OSI trata da conexão de sistemas abertos e segue os seguintes princípios:

- Deverá ser criada uma camada na qual é necessário um nível diferente de abstração;
- Cada nível deverá executar uma função bem definida;
- A função de cada camada deverá ser escolhida tendo em vista a definição de protocolos padronizados internacionalmente;
- Os limites da camada deverão ser escolhidos de forma a minimizar o fluxo de informações através das interfaces;
- O número de camadas deverá ser grande o bastante para comportar funções distintas que possam ser utilizadas simultaneamente em uma camada e pequena o bastante de forma que a arquitetura não se torne pesada.

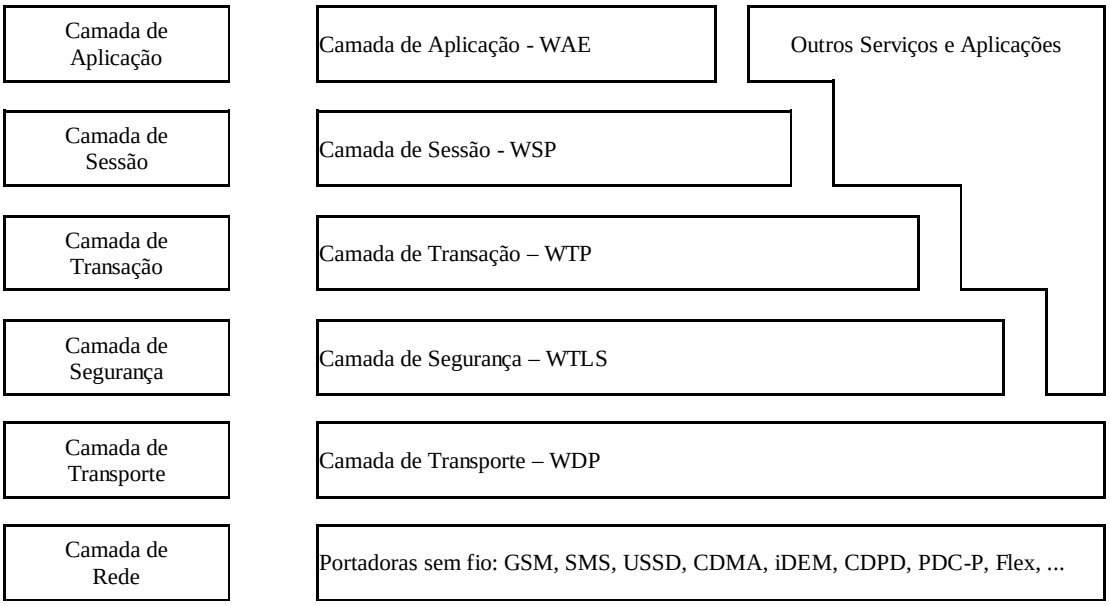

Figura 3.1 - Pilha de Protocolos WAP

Camadas do protocolo WAP (SHARMA, 2001, WATSON, 2000):

- *Wireless Application Environment – WAE* (Ambiente de Aplicação Sem Fio). Essa camada inclui: micronavegador, WML (linguagem de marcação sem fio), WML Script (linguagem de script de cliente), WTA (aplicação de telefonia sem fio) e suporte para texto e imagens;
- *Wireless Session Protocol – WSP* (Protocolo de Sessão Sem Fio). Destina-se a redes sem fio de pequena largura de banda e alta latência. e fornece funcionalidade HTTP 1.1;
- *Wireless Transactional Protocol – WTP* (Protocolo Transacional Sem Fio). Fornece serviço de transporte, permitindo transações confiáveis pergunta/resposta;
- *Wireless Transport Layer Security – WTLS* (Camada de Transporte Segura Sem Fio). Fornece privacidade, segurança de dados, autenticação e proteção contra ataque e serviços não-autorizados.
- *Wireless Datagram Protocol – WDP* (Protocolo de Datagrama Sem Fio). Camada de Transporte, suporta os protocolos: GSM, SMS, USSD, CDMA, iDEM, CDPD, PDC-P, Flex, entre outros.

3.4 Servidores e Gateway WAP

O navegador Web para recuperar e enviar informações conecta-se a servidores HTTP, essa mesma forma é adotada nos dispositivos móveis, que além de acessar servidores HTTP, podem acessar conteúdo através de um servidor WAP, pode ser analisada na figura a seguir (HEIJDEN e TAYLOR, 2000, FORTA et al, 2000).

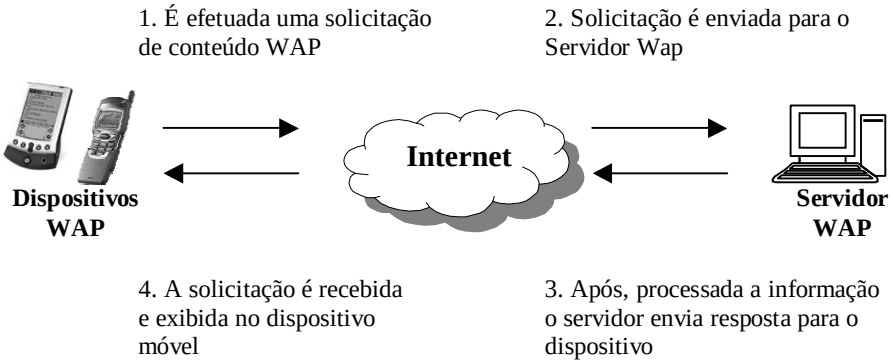

Figura 3.2 - Acesso às informações WAP

Para o dispositivo móvel acessar conteúdo HTTP, é necessário um servidor Gateway entre o dispositivo WAP e o servidor HTTP, que executará a função de transformação de HTTP para WLM e vice-versa, conforme mostra a figura 3.2.

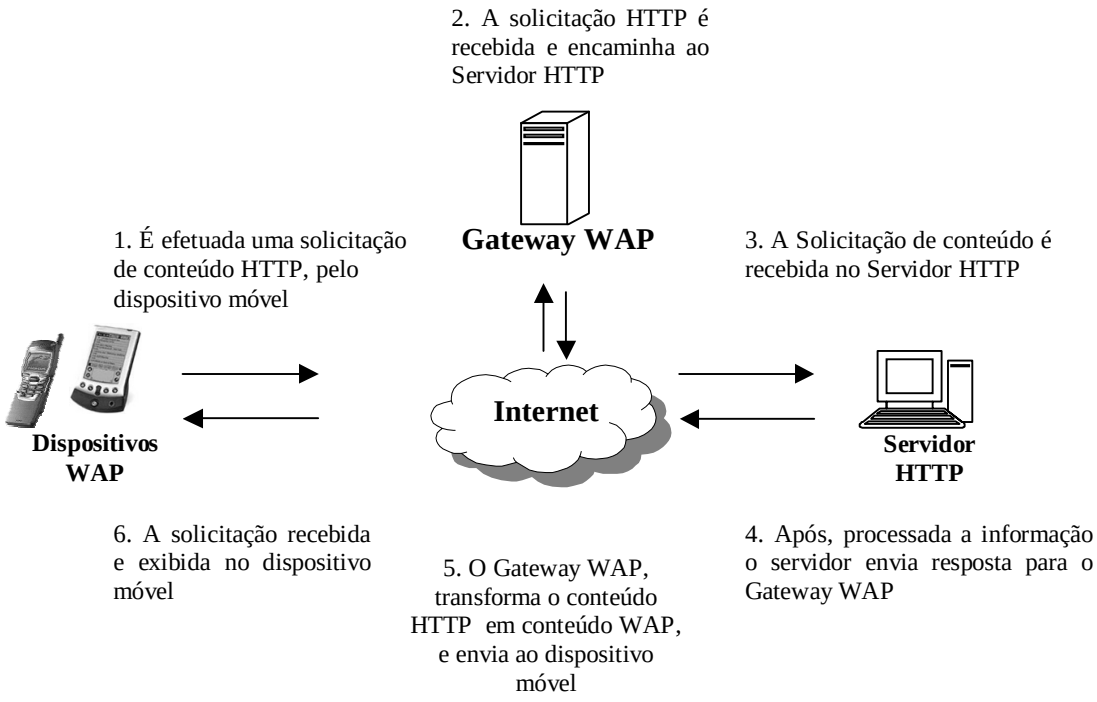

Figura 3.3 - Acesso às informações HTTP

3.5 Servidores de Aplicação WAP

Segundo MANN (1999), para facilitar a distribuição do conteúdo WML seria a necessária a utilização de um servidor HTTP. Assim como *www* tornou-se *host* para servidores HTTP, o *wap* passou a ser padrão para servidores de aplicativos WAP.

Exemplo: wap.nomedosite.com.br

Para que os aplicativos WAP possam ser visualizados corretamente nos dispositivos móveis, é necessário informar ao navegador que ele irá receber uma página WML e não HTML. Para isso são utilizadas extensões do tipo MIME, que serão incorporados no servidor.

3.5.1 Tipos MIME

Os tipos MIME necessários são:

Tipo do Conteúdo	Tipo MIME	Extensão
Código WML	text/vnd.wap.wml	wml
WML Compilado	application/vnd.wap.wmlc	wmlc
WML Script	text/vnd.wap.wmlscript	wmls
WML Script Compilado	application/vnd.wap.wmlscriptc	wmlsc
Imagem WML	imagem/vnd.wap.wbmp	wbmp

Tabela 3.1. Tipos de arquivos e tipos MIME

3.5.2 Configurando os tipos MIME nos servidores

Baseado em DIAS (2000b) e SETUBAL (2000), para configuração do tipo MIME nos servidores Apache HTTP Server, Microsoft IIS Server v4.0, Microsoft IIS Server v3.0 e versão anterior ao Microsoft Personal Web Server v4.0, segue:

Apache HTTP Server (anteriores a 1.3.4):

1. Edite o arquivo srm.conf.
2. Localize a seção AddType e adicione o seguinte ao arquivo:
    a. # MIME Types for WAP
    b. AddType text/vnd.wap.wml .wml
    c. AddType text/vnd.wap.wmlscript .wmls
    d. AddType image/vnd.wap.wbmp .wbmp
    e. AddType application/vnd.wap.wmlc .wmlc
    f. AddType application/vnd.wap.wmlscriptc .wmlsc
3. Salve o arquivo e reinicie o Apache HTTPd.

Microsoft IIS Server:

1. No console do servidor, abra o Management console ou a Intenet Service Manager Tool.

2. No Management console, defina os tipos MIME como válidos para todo o servidor ou apensas para diretórios separados.
3. Para adicionar um novo tipo MIME a um diretório específico, clique com o botão direito do mouse no diretório desejado e selecione Properties.
4. Selecione a guia HTTP headers.
5. Clique no botão File Types no canto inferior direito.
6. Selecione New Types e os seguintes valores de campo:
    a. Extensão associada: .wml
    b. Tipo de conteúdo (MIME): text/vnd.wap.wml
7. Clique em OK.
8. Repita as etapas 6 e 7 para cada tipo de MIME adicional.
9. Reinicie o sistema.

Microsoft Personal Web Server v.4.0:

1. Utilizando a ferramenta MetaEdit no kit de recursos do Microsoft IIS.
2. Abra o /MIMEMAP em /LM.
3. Selecione MimeMap.
4. Adicione a lista:
    a. wml, text/vnd.wap.wml
    b. wmls, text/vnd.wap.wmlscript
    c. wbmp, image/vnd.wap.wbmp .wbmp
    d. wmlc, application/vnd.wap.wmlc
    e. wmlsc, application/vnd.wap.wmlscriptc
    f. hdml, text/x-ndml
5. Clique em OK.
6. Reinicie o sistema.

IIS v3.0 ou versão anterior ou Personal Web Server v1.0

1. Execute o REGEDIT.EXE.
2. Abra a chave do registro:

HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\InetInfo\Parameters\MimeMap

3. Adicione novos valores a lista:
    a. text/vnd.wap.wml, wml,,5
    b. text/vnd.wap.wmlscript, wmls,,5
    c. image/vnd.wap.wbmp,,wbmp,,5
    d. application/vnd.wap.wmlc,wmlc,,5
    e. application/vnd.wap.wmlscript,wmlsc,,5
    f. text/x-hdml,,5
4. Reinicie o sistema.

3.6 Segurança

Um fator muito importante quando surge uma nova tecnologia é a segurança, barreira difícil de ser totalmente quebrada. E mesmo na época em que a comunicação se restringia somente a voz, a segurança sempre foi levada como um ponto inicial de credibilidade de bom serviço (DORNAN, 2001).

Pelo fato da tecnologia WAP estar diretamente ligada a internet, seus aplicativos tomam por base os caminhos usados na internet convencional e também os seus problemas ligados à segurança acompanham o processo.

Todo sistema seguro deve levar em consideração quatro pontos básicos: privacidade, integridade, autenticidade e não-repúdio (ALBERTIN ,2000, DENEGA, 2000).

- Privacidade: garantia que o conteúdo possa ser lido por intermédio de uma mensagem entre o emissor e o receptor;
- Integridade: garantia que o conteúdo não sofreu alteração alguma do envio ao recebimento da mensagem;
- Autenticidade: garantia que tanto o emissor como o receptor da mensagem são realmente quem dizem ser. São utilizados Certificados Digitais para garantir autenticidade;
- Não-repúdio: comprovante que o emissor enviou a mensagem e o receptor a recebeu, onde nenhuma das partes pode negar a participação na transação.

3.6.1 Segurança no WAP e na WEB

Os aplicativos Web utilizam a tecnologia SSL (Secure Sockets Layer), sendo a criptografia de chave pública o principal elemento para sua segurança. Na criptografia por chave pública, conforme visto na figura 3.4 (sendo a mesma utilizada anteriormente na seção 2.1.1 para melhor entendimento), o emissor codifica os dados utilizando uma chave privada e envia ao receptor uma chave para decodificar a mensagem, garantindo privacidade e integridade (DENEGA, 2000, TERRY et al, 1997).

O SSL foi desenvolvido para computadores pessoais com grande capacidade de processamento, maior largura de banda e baixa latência. Porém, a inclusão dos protocolos SSL nos dispositivos móveis, seria inviável, pois, os preços seriam elevados consideravelmente conforme afirma HJELM (2000).

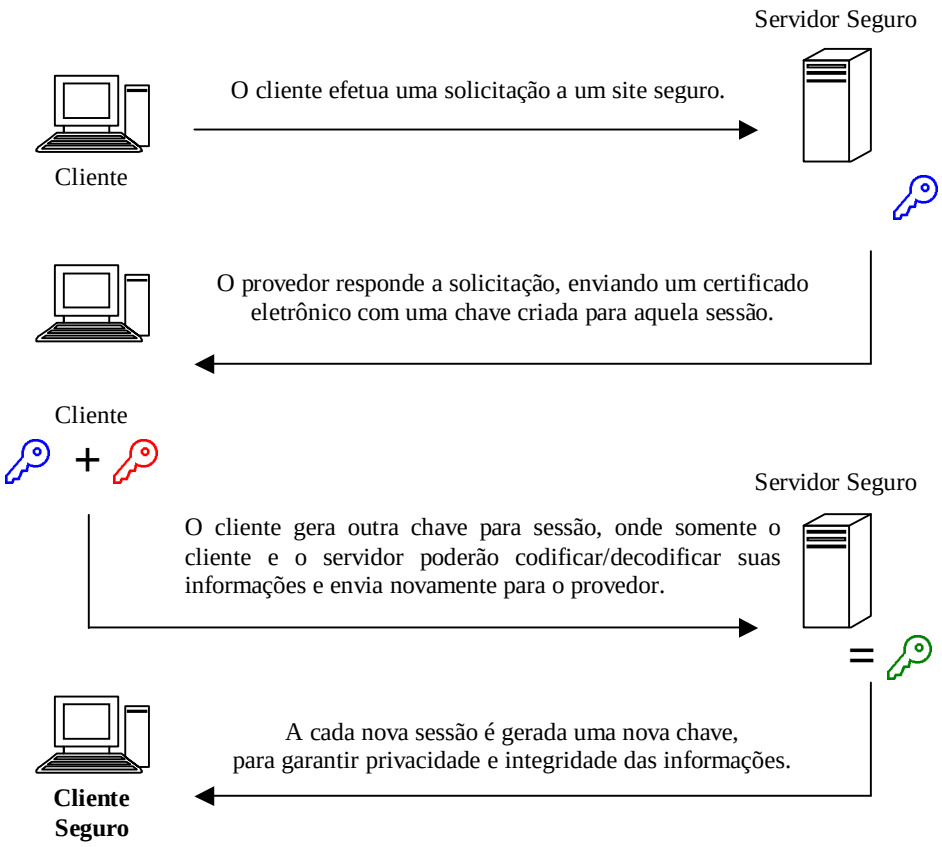

Figura 3.4 – Segurança no protocolo SSL

Os aplicativos WAP utilizam o protocolo WTLS (Wireless Transport Layer Security), para garantir a segurança nas transações em dispositivos sem fio. O protocolo WTLS se baseia na especificação TLS (Transport Layer Security) 1.0, que origina da especificação SSL 3.0. Oferecendo aos usuários: privacidade, integridade e proteção nos serviços de navegação (OLIVEIRA, 2000).

Ainda segundo DEMÉTRIO (2000), a WTLS agrega um bom nível de segurança, porém, atendendo as restrições impostas pelos dispositivos *wireless*, como baixa capacidade de armazenamento, menor largura de banda e alta latência.

A segurança no WAP se divide em duas partes distintas, conforme ilustra a figura 3.5, de um lado o gateway WAP utilizando o protocolo SSL para garantir uma comunicação segura com o servidor Web. E de outro, o dispositivo móvel trocando informações com o gateway através do protocolo WTLS.

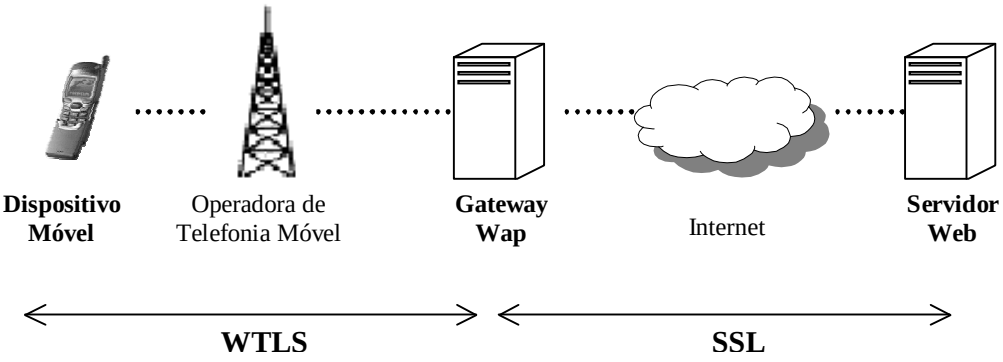

Figura 3.5 – Transações entre WTLS e SSL

DORNAN (2001) afirma que o principal problema da maioria dos sistemas sem fio é a falta de criptografia de ponta-a-ponta, de não serem capazes de cobrir uma conexão inteira, do dispositivo móvel ao servidor Web.

Comparado com as redes de telefonia fixa, RISCHPATER (2001) afirma que as redes sem fio garantem maior segurança, oferecendo sempre aos seus usuários algum tipo de criptografia. As redes *wireless* na maioria dos casos utilizam algum tipo de compartilhamento de freqüência, onde os dados são distribuídos de uma única fonte através de vários canais.

Sendo a captura e a remontagem de grande quantidade de pacotes, por parte de invasores, inviável. Pois, em redes de espectro de difusão seria necessário conhecimento interno de como seus sinais estão sendo distribuídos.

O Gateway Wap é o responsável pela transformação do protocolo WAP para o protocolo da Web. A duração do processo de conversão entre SSL e WTLS, é de apenas alguns milissegundos e ocorre juntamente com outras requisições, tornando uma conexão segura entre os protocolos (MANN, 1999).

Em contrapartida, FORTA et al (2000) adverte, que a vulnerabilidade nesse pouco espaço de tempo pode ser suficiente para um ataque externo e a concessão do gateway WAP. Como o gateway é o intermediário, sendo invadido por um *hacker* (cracker), os dispositivos móveis e a Web ficariam vulneráveis.

Algumas formas de ataque são:

- Ataques de curiosos (eavesdropping), que ocorre entre o dispositivo móvel e o gateway WAP;
- Ataque de intermediário (Man-in-the-Middle), ocorre no gateway WAP;
- Ataque entre o gateway WAP e o servidor Web;
- Ataque por repetição (Replay) da solicitação do gateway WAP.

DENEGA (2000) alerta sobre os cuidados que devem ser tomados no gateway WAP para suprir essa falha de segurança no modelo WAP:

- Utilização de um *Firewall*;
- Não manter conteúdo decodificado em meio secundário;
- Remoção do conteúdo original na memória volátil o mais breve possível;
- Restrição de acesso físico ao equipamento;
- Restrição de acesso administrativo ao *gateway* e diretórios internos;
- Utilização de padrões habituais de segurança.

Tanto a camada de transporte WTLS, quanto o SSL, não garantem autenticação nem o não-repúdio, tendo para isso utilizar mecanismos próprios, como fazem os aplicativos HTML.

Capítulo 4

WML

Wireless Markup Language (Linguagem de Marcação Sem Fio), é uma linguagem de marcação baseada na linguagem XML (Extensible Markup Language), baseada em tags. Utilizada para especificar a estrutura dos documentos que serão transmitidos nos dispositivos móveis como celulares e PDA (Assistentes Digitais Pessoais) (DIAS, 2000a, FROST, 2001).

Segundo RISCHPATER (2001), a WML foi projetada exclusivamente para:

- ambientes que possuem limitação de hardware;
- largura de banda estreita;
- recursos de entrada e saída limitados;
- telas restritas;
- conexões instáveis;
- baixa capacidade de processamento e armazenamento;

Considerando as afirmações de ARAUJO (2001), a WML é uma linguagem robusta, consistente e leve, atendendo as necessidades dos profissionais, sendo de fácil utilização. Outro ponto positivo, está em sua portabilidade, pois, roda em qualquer sistema operacional.

A WML oferece os seguintes recursos, MANN (1999):

- Suporte a Texto e Imagens: incluindo dicas de apresentação como quebras de linhas, formatação de textos e alinhamento. Embora não seja obrigado a suportar imagens, a WML oferece suporte para o formato WBMP (*Wireless Bitmap*);
- Entrada do usuário: o WML oferece suporte a entrada de textos, listas de opções e controles de tarefa;

- Mecanismos de navegação: suporte aos *links* ancorados e esquema de nomes URLs (padrão da Web). Permitindo ir de um *card* a outro, ou a um novo *deck*;
- Suporte a múltiplos idiomas e dialetos;
- Gerenciamento de estado e de contexto: permite ao usuário um amplo controle sobre as variáveis, onde as variáveis podem ser enviadas e substituídas no tempo de execução. Armazenando em *cache* as variáveis e arquivos WML, minimizando solicitações do servidor.

4.1 Cards e Decks

Deck (ou baralho) é a forma de indicar cada documento ou página WML que é transmitida para um dispositivo WAP. É formado basicamente pelo prólogo, *tags* WML <wml>, *tags* de cabeçalho <head>, outras tags da WML e *cards*.

Cards (ou cartas) significa uma unidade de interação com o usuário, como uma opção de menu ou tela de texto. Define a aparência de cada tela, informações de formatação, conteúdo exibível e instruções de funcionamento. Todo *card* começa e termina com a *tag* <card> (HOMMER et al, 2000).

Como mostra figura 4.1, os *cards* são agrupados dentro dos *decks* (SETUBAL, 2000), sendo a menor unidade do WML que o servidor Web pode enviar para um dispositivo (agente usuário ou UA – User Agent).

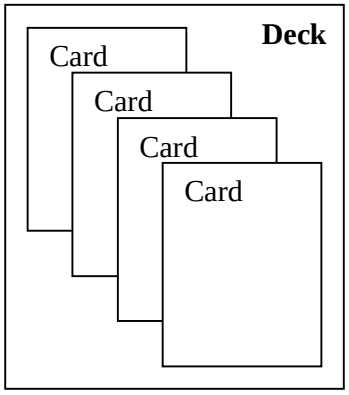

Figura 4.1 – Agrupamento de *cards* no *deck*.

Quando um micronavegador acessa um documento WML, primeiramente efetua a leitura de todo o conteúdo do *deck*, sendo assim, a navegação entre os cards é feita sem a necessidade de carregar novamente o mesmo *deck*, até uma nova solicitação (FROST, 2001).

4.2 Comandos WML

Os comando WML podem ser divididos em grupos conforme sua utilização (ARAUJO, 2001, FACUNTE, 2000, WATSON et al, 2000):

- Formatação de Texto
- Eventos
- Tarefas
- Entrada de Dados
- Deck / Card
- Variáveis
- Ancoras / Imagens / Timers

4.2.1 Formatação de Texto

<!-- texto **-->** → Inserção de comentários.

**** → Texto em negrito.

Sintaxe:

<big> → Texto em formato ampliado.

Sintaxe: <big>
</big>

**
** → Quebra de linha. É utilizado para o caso de se desejar colocar o texto ou uma imagem em uma nova linha.

Sintaxe:

**** → Texto com ênfase.

Sintaxe:

**** → Texto em itálico.

Sintaxe:

** ** → Define formatação do parágrafo ou linha em branco.

Sintaxe:

<small> → Texto em formato reduzido.

Sintaxe: <small>
</small>

**** → Texto com ênfase e negrito.

Sintaxe:

<table> → Cria uma tabela.

Sintaxe: <table columns=" " title=“ ” align=" ">
<tr>
<td></td>
</tr>
</table>

<u> → Texto sublinhado.

Sintaxe: <u>
</u>

<td> → Define as colunas da tabela.

Sintaxe: <td>

<tr> → Define as linhas da tabela.

Sintaxe: <td>

4.2.2 Eventos

<do> → Realiza uma ação. Pode estar associado aos elementos: <go>, <prev>, <refresh> e <noop>.

Sintaxe:<do type=tipo label=título name identificador optional = true/false>
</do>

<onevent> → Liga uma tarefa a um evento particular para o elemento imediatamente incluído. Pode estar associado aos elementos: <go>, <prev>, <refresh> e <noop>.

Sintaxe: <onevent type=""> tarefas>
</onevent>

<ontimer> → Pode ser incluído dentro dos elementos *card* e *template,* sendo equivalente ao elemento <go>, porém, acessará o url especificado quando o *timer* expirar.

<postfield> → Envia valores de variáveis para o servidor. É utilizado em conjunto com o elemento <go>.

Sintaxe: <postfield name=" " value=" " />

4.2.3 Tarefas

<go> → Direciona-se a uma nova URL, um *card* ou *deck.*

Sintaxe: <go href=" " sendreferer=" " method=" " accept-charset=" " />

<prev> → Retorna a URL anterior ou *deck*.

Sintaxe: <prev>
</prev>

<refresh> → Atualiza cartão.

Sintaxe: <refresh>
</refresh>

<noop> → Elemento que instrui o dispositivo para não efetuar nenhuma operação. Utilizado para desconsiderar uma chamada do elemento <go>.

Sintaxe: <noop/>

4.2.4 Entrada de Dados

<fieldset> → Permite o agrupamento de campos relacionados e texto.

Sintaxe: <fieldset title=" ">
</fieldset>

<input> → Permite a entrada de valores e designa para uma variável especificada. Pode conter um conteúdo preestabelecido.

Sintaxe: <input name=" " title=" " type=" " value=" " default=" " format=" " emptyok=" " size=" " maxlength=" " tabindex=" "/>

<optgroup> → Permite agrupar múltiplos elementos option relacionados em uma hierarquia.

Sintaxe: <optgroup title=" ">
</optgroup>

<option> → especifica uma única opção de escolha em um elemento selecionado.

Sintaxe: <option value=" " title=" " onpick=" ">
</option>

<select> → Especifica uma lista de opções que o usuário pode selecionar (uma ou mais opções). Está relacionado com o elemento <option>.

Sintaxe: <select name="" title=" ">
</select>

4.2.5 Deck / Card

<access> → Especifica a informação de controle do acesso para um pacote WML. Não deve ser utilizado mais de um elemento <access> dentro de um pacote.

Sintaxe: <access domain="" path=""/>

<card> → Representa uma tela no dispositivo (exibe no máximo um *card* por tela). Um *deck* é composto por um ou mais elementos *cards*.

Sintaxe: <card id=" " title=" " newcontext=" " ordered=" " ontimer=" " onenterforward=" " onenterbackward=" ">
</card>

<head> → É o cabeçalho, onde se encontram as informações inclusive meta-data e o controle de acesso definido pelo tag <access>.

Sintaxe: <head>
</head>

<meta> → Especifica um *meta-information* com nomes de propriedades e valores. Nem todos os dispositivos WAP suportam meta.

Sintaxe: <meta http-equiv=" " name=" " forua=" " content=" " scheme=" "/>

<template> → Define um modelo geral de operação para todos os *cards* no dispositivo. Pode-se sobrepor essas características para um *card* em especial.

Sintaxe: <template onenterfoward=" " onenterbackward=" " ontimer=" ">
</template>

<wml> → *Tag* que define o início e o fim de um *deck*. Elemento <wml> especifica um dispositivo do WML.

Sintaxe: <wml xml:lang=" ">
<card>
</card>
</wml>

4.2.6 Variáveis

<setvar> → Define o valor a uma variável, quando é executado um elemento <go>, <prev> ou <refresh>.

Sintaxe: <setvar name=" " value=" "/>

4.2.7 Ancoras / Imagens / Timers

<a> → Especifica um hiperlink, porém, usada para definir o elemento <go>, exigindo um URL.

Sintaxe:

<archor> → Especifica um hiperlink com qualquer texto formatado ou imagem.

Sintaxe: <anchor title=" ">
</anchor>

<img> → O elemento <img> indica que será inserido uma imagem ou ícone no documento WML. Os dispositivos atuais suportam apenas imagens do tipo *wbmp*.

Sintaxe:

<timer> → Efetua uma ação após um período de tempo determinado.

Sintaxe: <timer name=" " value=valor />

4.3 Acentuação e Caracteres especiais

Para utilização de caracteres especiais e/ou acentuados nos sites WML, devemos utilizar alguns códigos, esses códigos relativos são referentes à ISO 8859-1 (ARAUJO, 2001, DEMÉTRIO, 2000).

Exemplo_1: para escrever Pizza Média, deve-se escrever 'Pizza M**é**dia', conforme código da tabela 4.1.

Exemplo_2: para escrever o símbolo > (maior que), deve-se escrever **>** , conforme tabela 4.2.

Escrita	Código	Função	Escrita	Código	Função
		espaço	×	×	×
¡	¡	¡	Ø	Ø	Ø
¢	¢	¢	Ù	Ù	Ù
£	£	£	Ú	Ú	Ú
¤	¤	¤	Û	Û	Û
¥	¥	¥	Ü	Ü	Ü
¦	¦	¦	Ý	Ý	Ý
§	§	§	&Thorn;	Þ	Þ
¨	¨	¨	ß	ß	ß
©	©	©	à	à	à
¬	¬	¬	á	á	á
®	®	®	â	â	â
°	°	°	ã	ã	ã
±	±	±	ä	ä	ä
´	´	´	å	å	å
¼	¼	¼	æ	æ	æ
½	½	½	ç	ç	ç
¾	¾	¾	è	è	è
¿	¿	¿	é	é	é
À	À	À	ê	ê	ê
Á	Á	Á	ë	ë	ë
Â	Â	Â	ì	ì	ì
Ã	Ã	Ã	í	í	í
Ä	Ä	Ä	î	î	î
Å	Å	Å	ï	ï	ï
Æ	Æ	Æ	ñ	ñ	ñ
Ç	Ç	Ç	ò	ò	ò
È	È	È	ó	ó	ó
É	É	É	ô	ô	ô
Ê	Ê	Ê	õ	õ	õ
Ë	Ë	Ë	ö	ö	ö
Ì	Ì	Ì	÷	÷	÷
Í	Í	Í	ø	ø	ø
Î	Î	Î	ù	ù	ù
Ï	Ï	Ï	ú	ú	ú
Ñ	Ñ	Ñ	û	û	û
Ò	Ò	Ò	ü	ü	ü
Ó	Ó	Ó	ý	ý	ý

Ô	Ô	Ô	þ	þ	þ
Õ	Õ	Õ	ÿ	ÿ	Ÿ
Ö	Ö	Ö			

Tabela 4.1 – Acentuação e Caracteres Especiais

Símbolo	**Escrita**
<	< (menor que)
>	> (maior que)
Hífen	
´	' (apóstrofo)
"	&quote; (aspas)
&	& (ampersand)
$	$$ (dólar)
Espaço Simples	

Tabela 4.2 – Caracteres Especiais

4.4 WML em linguagem Dinâmica

O conteúdo WML, como HTML é criado de duas formas: arquivos estáticos e arquivos dinâmicos (BATTISTI, 2000). Os estáticos são armazenados no servidor e enviados aos clientes, sem o processamento das informações (simplesmente são enviados os arquivos). Já os arquivos dinâmicos, as informações são processadas no servidor e enviados os resultados ao solicitante (dispositivo móvel ou *site* da Web).

Suas principais funções são: interação com banco de dados, processamento de informações e integração por e-mail (FORTA et al, 2000). Para os arquivos estáticos a conversão é diretamente efetuada pelo servidor WAP ou HTTP, quando é lido uma extensão wml. Já para arquivos dinâmicos, o agente usuário para reconhecer que um arquivo recebido é um conteúdo WAP, o arquivo deve ser especificado pelo tipo MIME. A configuração do tipo MIME para os principais editores de linguagem dinâmica devem seguir as seguintes sintaxes para especificação do conteúdo WAP:

- Microsoft ASP

  Sintaxe: <% Response.ContentType="text/vnd.wap.wml" %>

- JSP (Java Server Pages - SUN)

  Sintaxe: <%@ page.contentType="text/vnd.wap.wml" %>

- Perl

  Sintaxe: print "Content-type: text/vnd.wap.wml\n\n";

- PHP

  Sintaxe: <?php ("Content-Type: text/vnd.wap.wml"); ?>

4.5 WMLScript

WMLScript (Wireless Markup Language Script) é uma linguagem de script cliente orientada a objetos, utilizada para suprir algumas deficiências deixadas pelo WML. Como a criação de páginas dinâmicas, dando maior velocidade ao processo, pois utiliza bibliotecas que são armazenadas no próprio *browser* do dispositivo WAP.

Baseada na ECMAScript (European Computer Manufacturers Association – Associação Européia de Fabricantes de Computador). Sua compilação é executada no gateway WAP em formato binário para fornecer processamento aos clientes móveis. É case-sensitive (diferencia caixa alta de caixa baixa). WMLScript deve permanecer em arquivo separado e não com *tags* na própria página como o JavaScript (FROST, 2001).

Aplicações onde podem ser incorporado o WMLScript (DIAS, 2000a):

- Validação de entrada do usuário;
- Interação com aplicativos no próprio dispositivo ou externo ao navegador;
- Processamento local das informações através das bibliotecas, agilizando as tarefas onde seria necessário retornar ao servidor para buscar respostas ou confirmações;
- Desenvolvimento de aplicativos mais ricos em recursos;
- Realização de operações matemáticas, manipulação de string de texto efetuar redirecionamentos.

4.5.1 Bibliotecas WMLScript

A WMLScript possui 6 bibliotecas (ARAUJO, 2001, FACUNTE, 2000, WATSON et al, 2000), são elas:

- Lang;

- Float;
- String;
- URL;
- WMLBrowser;
- Dialogs.

Biblioteca Lang

A biblioteca Lang contém um conjunto de funções que são à base da linguagem WMLScript. Semelhante a biblioteca utilizada pelo JavaScript. Exemplo: gerar um número aleatório.

Função	**Descrição**
abort(string)	Termina interpretação e retorna string.
abs(num)	Retorna o valor absoluto de num.
characterSet()	Retorna o valor do conjunto de caracteres IANA (Internet Assigned Numbers Authory).
exit(valor)	Termina interpretação e retorna valor.
float()	Retorna verdadeiro se a casa decimal do valor é suportada.
isFloat(string)	Retorna verdadeiro se string for convertida para um número decimal utilizando parseFloat().
isInt(string)	Retorna verdadeiro se string for convertida para número inteiro utilizando parseInt().
max(num1, num2)	Retorna o valor máximo de num1 ou num2.
maxInt()	Retorna o valor máximo suportado.
min(num1, num2)	Retorna o valor mínimo de num1 ou num2.
minInt()	Retorna o valor mínimo suportado.
parseFloat(string)	Retorna o ponto flutuante equivalente de string.
parseInt(string)	Retorna o valor inteiro equivalente de string.
Random(int)	Retorna um inteiro randômico entre zero e int.
seed(int)	Inicializa o gerador de números randômicos utilizando int.

Tabela 4.3 – Biblioteca Lang

Biblioteca Float

Possui um conjunto de funções aritméticas com capacidade de manipular números com ponto flutuante. Sendo opcional, utilizado por dispositivos que suportam números decimais. Exemplo: retornar a raiz quadrada de um número.

Função	**Descrição**
cell(num)	Retorna o menor inteiro não menor que num.
floor(num)	Retorna o maior inteiro não maior que num.
int(num)	Retorna a parte inteira de num.
maxFloat()	Retorna o maior valor com ponto flutuante aceito.
minFloat()	Retorna o menor valor com ponto flutuante aceito.
pow(num1,num2)	Retorna o valor de num1 elevado à potência de num2.
Round(num)	Retorna o inteiro mais próximo de num.
sqrt(num)	Retorna a raiz quadrada de num.

Tabela 4.4 – Biblioteca Float

Biblioteca String

A biblioteca String possui um conjunto de funções de string. Uma string em WMLScript é basicamente uma matriz (array) de caracteres. Onde o primeiro elemento possui deslocamento 0. Exemplo: retornar o número de caracteres existente dentro de uma string.

Função	Descrição
charAt(string, num)	Retorna o caractere string da posição num.
compare(string1,string2)	Compara duas strings, retorna 1 se a 1ª string é maior que a 2ª, 0 se as duas tem o mesmo valor e -1 se a 1ª é menor que a 2ª.
elementAt(string, num, stringSep)	Retorna o enésimo elemento de string separado por stringSep.
elements(string, stringSep)	Retorna o número total de elementos contidos na string especificada.
find(string, subString)	Retorna a posição da subString, caso seja encontrada a string.
format(stringFmt, valor)	Converte valor em uma string usando a string de formatação strFmt.
InsertAt(string, stringElem, num,stringSep)	Retorna stringElem em string no enésimo elemento separado por stringSep.
IsEmpty(string)	Retorna verdadeiro se a string estiver vazia.
Length(string)	Retorna o comprimento da string.
removeAt(string, num, stringSep)	Retorna uma string com o elemento especificado removido.
replace(string, stringAnt, stringNova)	Substitui todas as stringAnt por stringNova.
replaceAt(string, strElem, num, stringSep)	Substitui o elemento especificado em uma string por um novo elemento.
sqeeze(string)	Remove todos espaços em branco da string.
subString(str, Inic, Tam)	Cria uma nova string a partir da str especificada, usando Inic como posição inicial e Tam como tamanho desejado.
toString(valor)	Retorna um valor numa string.
trim(string)	Remove todos espaços em branco da direita e esquerda da string.

Tabela 4.5 – Biblioteca String

Biblioteca URL

Possui um conjunto de funções que são utilizadas no tratamento das URLs absolutas e URLs relativas. Exemplo: retornar o URL solicitado no arquivo atual.

Função	Descrição
escapeString(string)	Substitui os caracteres especiais contidos em uma string por equivalentes hexadecimais.
GetBase()	Retorna o URL absoluto do arquivo WMLScript atual.
getFragmente(string)	Retorna o fragmento do URL de string.
GetHost(string)	Retorna o host especificado do URL de string.
getParameters(string)	Retorna o parâmetro do URL de string.
GetPath(string)	Retorna o caminho do URL de string.
getPort(string)	Retorna o número da porta do servidor do URL de string.
getQuery(string)	Retorna a parte da consulta especificada na URL.
getReferer()	Retorna o URL que chamou o arquivo atual.
getScheme(string)	Retorna o esquema do protocolo internet de string.
isValid(string)	Retorna verdadeiro se string é um URL válido.
loadString(stringURL, stringTipoC)	Retorna o conteúdo encontrado no URL.
resolve(stringBase, stringEmb)	Retorna um URL combinando stringBase e stringEmb.
unescapeString(string)	Retorna o URL onde uma seqüência de escape foi substituída pelos caracteres apropriados.

Tabela 4.6 – Biblioteca URL

Biblioteca WMLBrowser

A biblioteca WMLBrowser possui funções onde o WMLScript possa acessar o contexto associado ao WML. Não podendo ter nenhum efeito colateral e devem retornar valores inválidos em casos onde o sistema não suporte o browser WML ou o interpretador WMLScript não for chamado pelo navegador WML. Exemplo: atualizar a interface com o usuário.

Função	Descrição
getVar(string)	Retorna o valor da variável string no contexto do browser atual.
setVar(string, valor)	Configura o valor da variável string para valor.
go(stringURL)	Carrega o conteúdo de stringURL.
Prev()	Retorna o browser para o cartão anterior.
refresh()	Atualiza a interface com o usuário.
newContext()	Limpa o contexto do agente de usuário WLM atual.
getCurrentCard()	Retorna o menor URL possível da carta do agente de usuário WML atual.

Tabela 4.7 – Biblioteca Browser

Biblioteca Dialogs

Esta biblioteca possui um conjunto de funções típicas para a interface (diálogos) com o usuário. Exemplo: enviar texto de mensagens para o dispositivo do usuário.

Função	**Descrição**
prompt(string, stringPad)	Exibe string e solicita uma entrada de dados usando stringPad como valor padrão.
Confirm(string, stringOk, stringCanc)	Exibe string, stringOk, stringCanc e retornará verdadeiro se stringOk for selecionada.
Alert(string)	Exibe string e aguarda confirmação.

Tabela 4.8 – Biblioteca Dialogs

4.6 Inserção de Imagem

Para utilização de imagens nas páginas WML, é utilizado a tag <img> (conforme ilustração Tabela 4.9) e deve seguir a seguinte sintaxe (DEMÉTRIO, 2000):

****

alt → caso a imagem não seja localizada ou o dispositivo não suportar imagens, será exibido no dispositivo o texto desejado.

src → especifica a URL da imagem a se exibida.

localsrc → lista de ícones armazenados na memória ROM do dispositivo móvel.

align → alinhamento da imagem em relação ao texto, podendo assumir três localizações na tela do dispositivo: top (topo), middle (meio) e bottom (base).

```
<card id="principal" ontimer="#tipoPizza">
  <timer value="30"/>
   <p align="center"><em>Pizzaria WAP</em></p>
   <p align="center">
  <img alt="Pizza" src="pizza.wbmp" align="top"/>
   <br/> (048) 224-9000 <br/>
        R. Bocaiúva, 1000 <br/>
        Florian&#243;polis-SC <br/>
   </p>
</card>
```

Tabela 4.9 – Inserção de Imagens

4.7 Temporizador

A principal função do temporizador é a atualização de informações em intervalos de tempo regulares, sem intervenção do usuário, garantindo sempre dados novos, permitindo considerar aplicativos em tempo real (AREHART et al, 2000). Utilizado na maioria dos casos na exibição de tela de abertura de uma página WAP ou em anúncios, dando dinamismo na abertura dos cartões (*cards*).

Para execução dessa tarefa é utilizado o elemento <timer>, que aguardará por um certo período de tempo uma determinada ação, e após este período, será iniciado automaticamente o processo, conforme mostra Tabela 4.9. O intervalo de execução no temporizador é dado em décimos de segundos (1/10), a partir da abertura do *card* (independente do tempo de abertura da imagem), tomando o cuidado para não definir o valor zero. Outra forma importante para utilização do temporizador é no tratamento de erro, onde é informado ao usuário o erro cometido, redirecionado após um período de tempo ao *card* anterior ao erro (ARAUJO, 2000).

A declaração do elemento <timer> segue a seguinte sintaxe:

<timer name="temporizador" value="50"/>

name → especifica o nome da variável que será definida com o valor do *timer*, opcional.

value → especifica a duração do *timer* e é atribuído ao atributo *name*, obrigatório. Neste exemplo a duração do timer seria de 5 segundos.

4.8 Imagens Animadas

Baixa capacidade de processamento, largura de banda estreita e limitação de hardware, são fatores que atualmente restringem a utilização de imagens animadas. O formato wbmp não suporta animação como *gifs* animadas utilizadas em páginas da Web. Porém, uma alternativa seria a criação de animações semelhantes aos desenhos animados (onde são criadas várias telas ou *cards* que serão colocadas em seqüência para dar a sensação de movimento, conforme figura 4.2), utilizando o elemento <timer>.

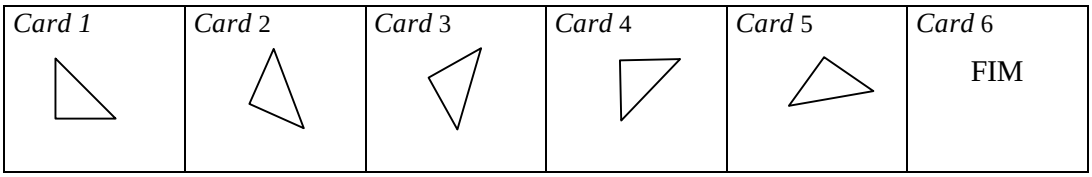

Figura 4.2 – Animação WAP

Os códigos referentes às animações da figura 4.2 seriam os seguintes (devem ser colocados os prólogos e a tag <wml>):

<!--Primeiro Cartão --> <card id="cartao1" ontime="#cartao2"> <timer value="10" /> </card>	<!--Quarto Cartão --> <card id="cartao4" ontime="#cartao5"> <timer value="10" /> </card>
<!--Segundo Cartão --> <card id="cartao2" ontime="#cartao3"> <timer value="10" /> </card>	<!--Quinto Cartão --> <card id="cartao5" ontime="#cartao6"> <timer value="10" /> </card>
<!--Terceiro Cartão --> <card id="cartao3" ontime="#cartao4"> <timer value="10" /> </card>	<!--Último Cartão --> <card id="cartao6" title="Fim"> THE END </card>

Tabela 4.10 – Cartões de Animação

FORTA (2000) adverte sobre a utilização de imagens animadas, pelo fato que nem todos os dispositivos WAP aceitam imagens e as telas possuem dimensões diferentes, onde os resultados poderão sofrer alterações. Outro problema que deve ser levado em consideração é o armazenamento em *cache*, onde cada quadro é adicionado ao histórico do dispositivo móvel, podendo ocorrer erros em função da dificuldade da limpeza posterior do *cache*.

4.9 Utilização do Cache

Após ser carregada no dispositivo móvel, a imagem passa a ser armazenada no *cache* por um período de tempo, assim como é executado em HTML, podendo ser utilizada em outros locais. Portanto, ao ser efetuada a requisição do mesmo código (por exemplo, uma imagem), primeiramente serão acessadas as informações contidas no *cache* do dispositivo, reduzindo as solicitações por parte do servidor.

Nos casos onde as informações sofrem alterações constantes os dados sempre estariam desatualizados em virtude do período em *cache* configurado pelo dispositivo. Nestes casos poderia ser utilizado o elemento <refresh> em determinada URL ou informar ao dispositivo que não será necessário armazenar determinada informação na sua memória. Para tanto, seria necessária a criação de cabeçalhos HTTP usando ASP, PHP, Perl ou qualquer outra linguagem compatível com o servidor (CHASE, 2000, MELONI, 2000, WEISSINGER,2000). As linhas a seguir seriam acrescentadas no cabeçalho HTTP para eliminação do *cache*:

Pragma: no-cache
Expires: Mon, 07 Jan 2002 0:00:00 GMT
Cache-Control: no-cache, must-revalidate
Pragma: no-cache

Capítulo 5

FERRAMENTAS PARA IMPLEMENTAÇÃO DE SITE DE M-COMMERCE

5.1 Editores para Linguagem WML

Os principais editores para linguagem WML são:

a) Adobe GoLive

O Adobe GoLive é um software para criação de páginas Web, em sua nova versão 6.0 oferece recursos num ambiente completo e integrado para criação de páginas compatíveis com WAP e iMode. Oferece suporte para as plataformas Windows e Macintosh, geração de código ASP e JSP.

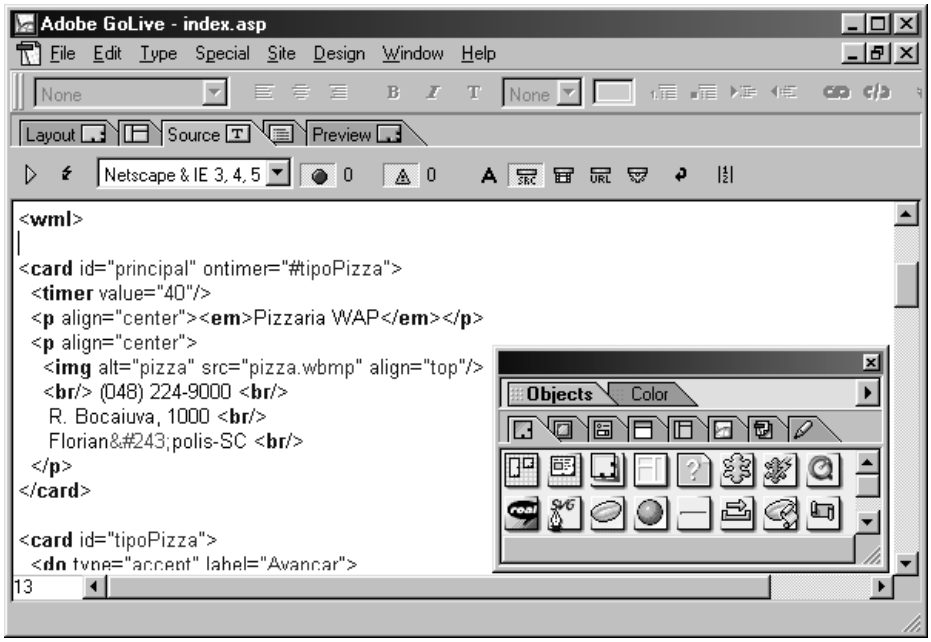

Figura 5.1 – Adobe GoLive

b) Allaire HomeSite

Considerado pelos desenvolvedores da Web como o melhor editor HTML do mercado, o Allaire Home Site oferece suporte integrado a linguagem WML. Um ponto negativo é o fato de não possuir um emulador WML, sendo necessário testar os códigos em outros programas.

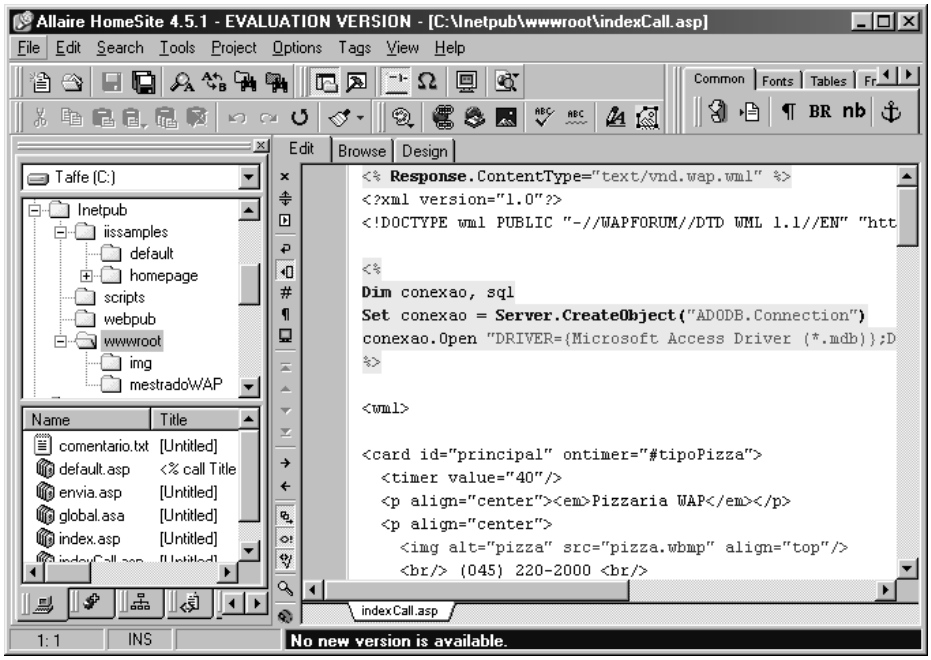

Figura 5.2 – Allaire HomeSite

c) Inetis DotWap

O Inetis DotWap é uma ferramenta *freeware*, para construção de páginas WAP. Permite a construção de sites de forma simples e rápida, sem conhecimentos específicos na área de programação WAP. Não possui flexibilidade para trabalhos de grande porte (utilizando linguagens dinâmicas como ASP, Perl, PHP, entre outras).

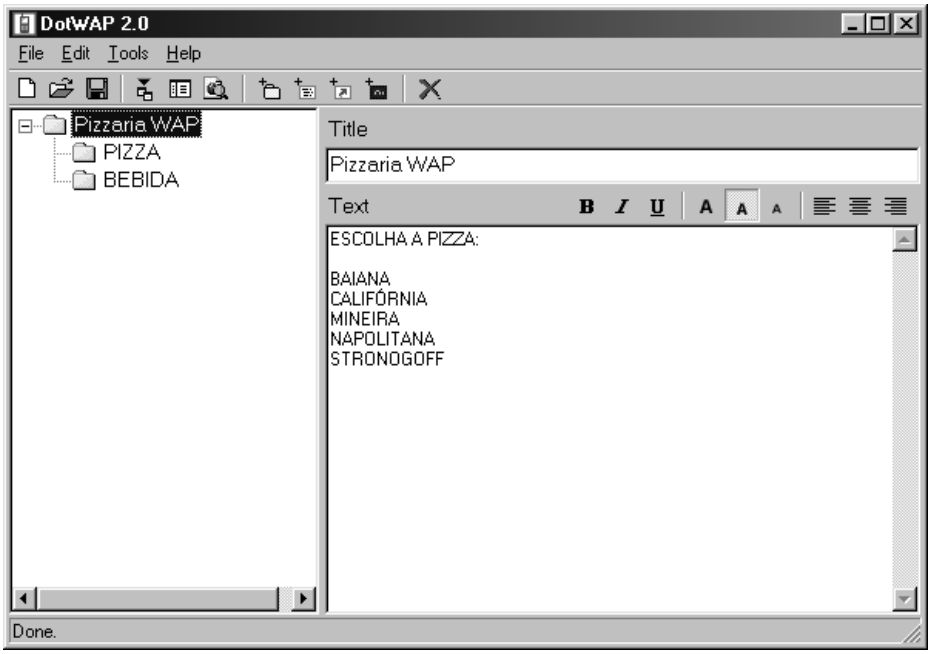

Figura 5.3 – Inetis DotWAP

d) Macromedia Dremweaver

O Dremweaver é um dos softwares de layout de página gráfica interativa mais utilizados para desenvolvimento de páginas Web. Apesar de não oferecer suporte direto a WML, em parceria com a NOKIA, lançou objetos *add-on* para o Dremweaver 3, tornado-se uma ferramenta que oferece suporte para criação e gerenciamento de *cards* e *decks.* Utiliza por padrão aparelhos celulares da NOKIA, mas apresenta simulação de outros aparelhos.

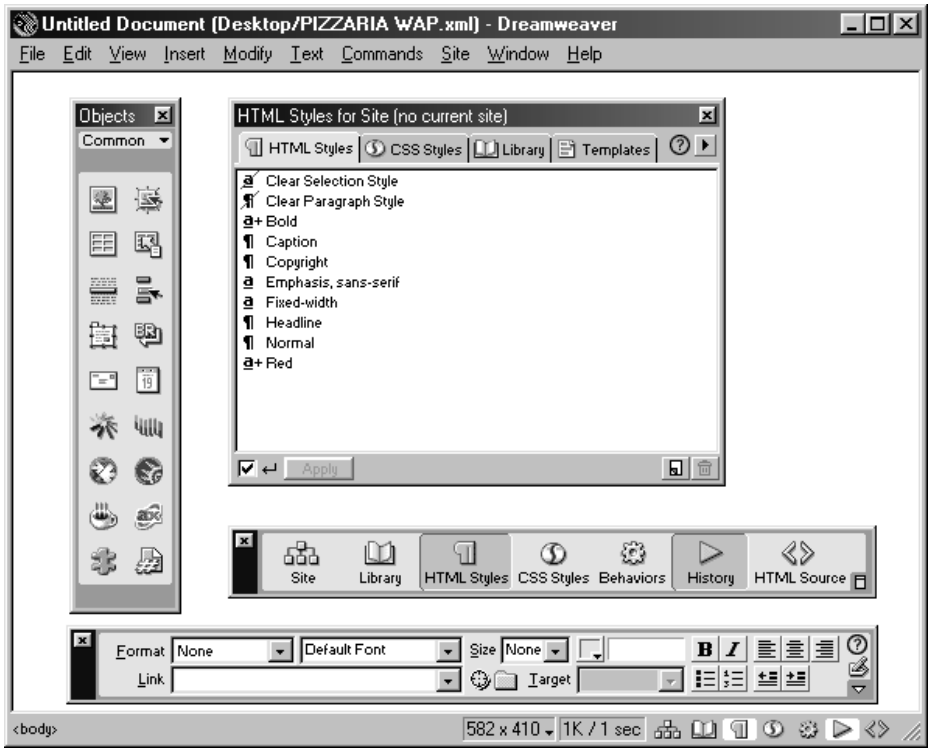

Figura 5.4 – Macromedia Dreamweaver

e) Rasquares Wap Pro

O Wap Pro é um dos principais e mais completos programas para edição de arquivos WML existentes no mercado, rodando em várias plataformas como: Windows 95 e 98, Windows NT e Windows 2000. Oferece recurso para edição de código de WML dinâmico criado com o ASP, Cold Fusion, JSP, Perl, PHP, WMLScript entre outros. Possui várias ferramentas que auxiliam o desenvolvedor de forma interativa na construção do código WML e um sistema de conversão automática de imagens do formato BMP ao formato WBMP.

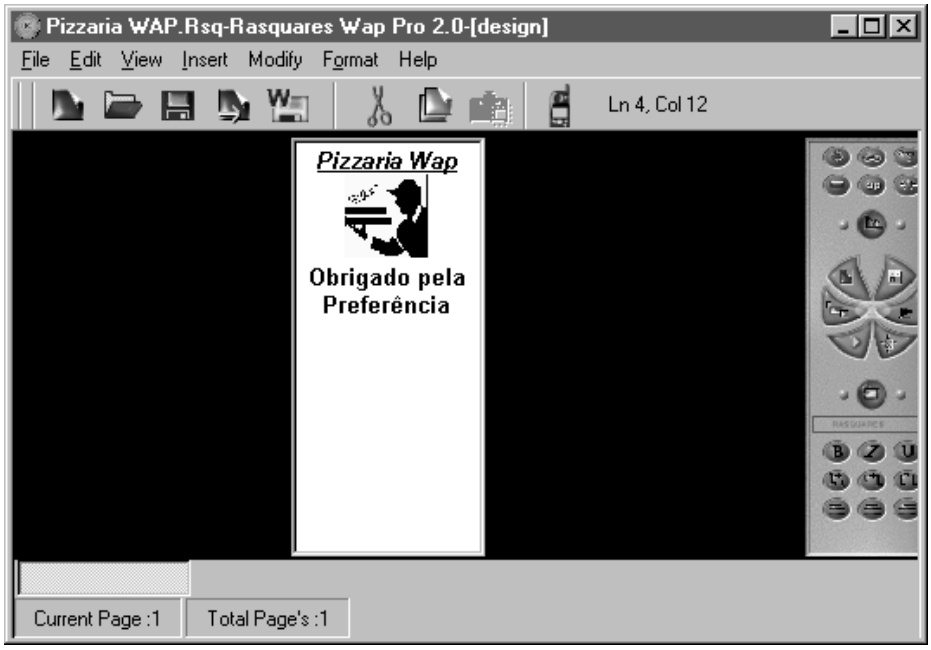

Figura 5.5 – Rasquares Wap Pro

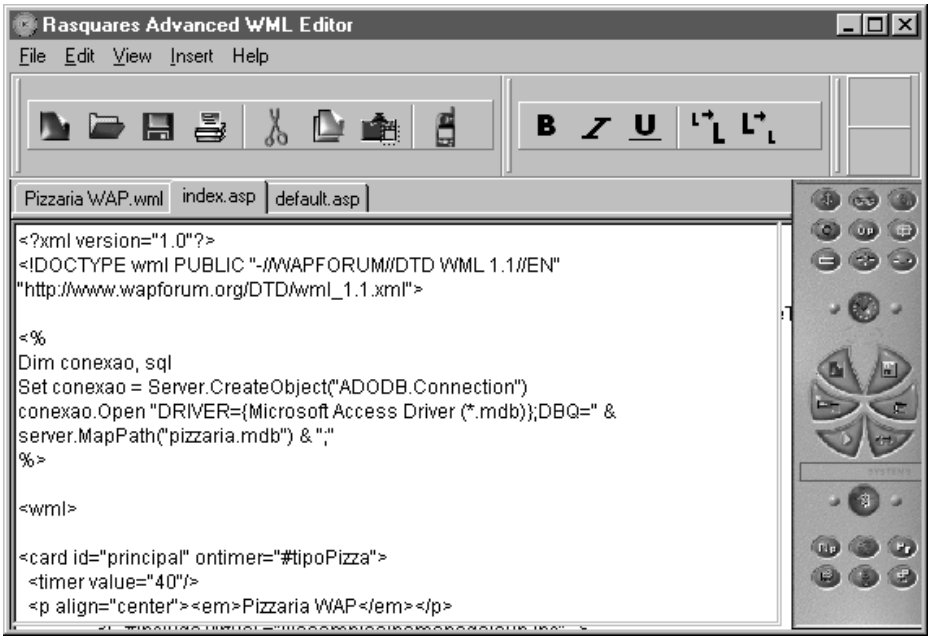

Figura 5.6 – Rasquares Wap Pro Editor

f) WapTop EasyPad WapTor

O WapTor é um editor para desenvolvimento de páginas WML, *freeware* que roda nos sistemas operacionais Windows 95/98, Windows NT e Windows 2000. Possui *layout* simples, sendo de fácil manuseio até mesmo para usuário iniciantes. Com o WapTor é possível criar e editar (várias páginas ao mesmo tempo) arquivos WML e visualiza-lo em seu simulador. Oferece suporte *on-line* para todas as *tags* WML.

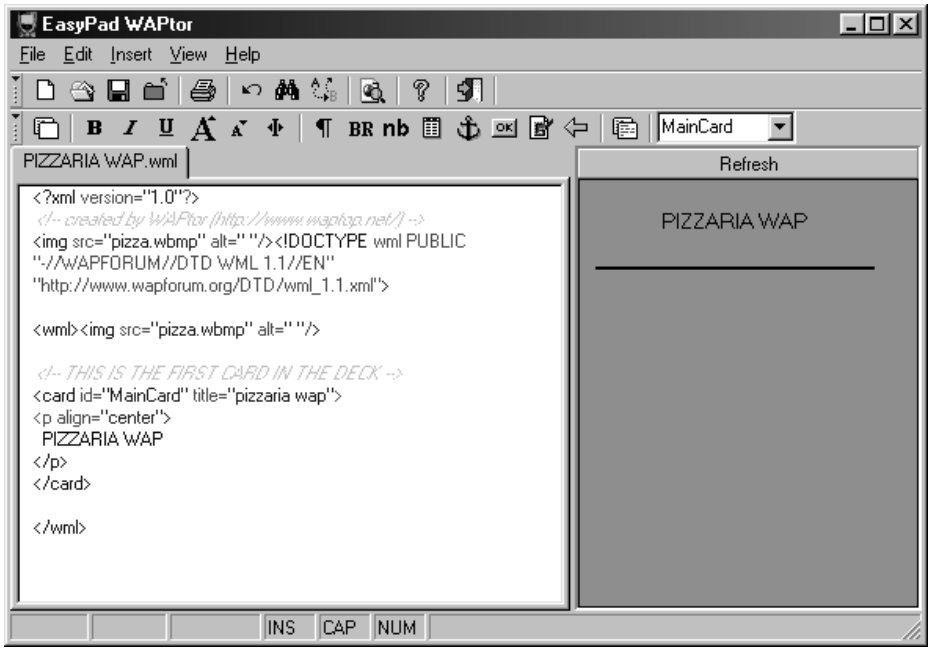

Figura 5.7 – WapTop EasyPad WapTor

5.2 Editores de Imagens para WML

Existem casos, em que uma imagem expressa tanto quanto muitas palavras e podendo ocupar menos espaço na tela do dispositivo móvel. (MANN, 1999).

Sendo sempre necessária uma nova solicitação de rede sempre que uma imagem é exibida, aumentando o tempo para a abertura de um card. Por isso, as imagens devem ser usadas com moderação e nos casos em que realmente ocupem menos espaço que sua forma em texto (RISCHPATER, 2001).

O formato de imagem adotado pelo WAP Fórum para dispositivos móveis é o WBMP (Wireless Bitmap Format). Onde, os gateways compatíveis com o WAP deverão reconhecer o WBMP como um tipo MIME (MANN, 1999).

As imagens quando enviadas aos dispositivos, são enviadas num único pacote, pois, os dispositivos atuais não suportam a remontagem de pacotes. Outro fator que deve ser levado em consideração é referente ao tamanho físico da tela, como a maioria dos dispositivos atuais possuem sistema de rolagem vertical (e não horizontal), as imagens podem ser cortadas. O comprimento das telas é de aproximadamente 40 pixels de largura (FORTA et al, 2000).

Segundo MANN (1999), pode ser utilizada uma lista de ícones, que estão armazenadas na ROM local do próprio dispositivo, através do atributo *localsrc*. Agilizando o processo, evitando uma operação de busca de dados adicionais.

Os principais editores de imagens são:

a) Butterfly

O Butterfly é um programa *shareware*, permite a conversão de imagens desenvolvidas no Adobe Photoshop, Macromedia Fireworks e Paint Shop Pro para o formato WBMP. É uma ferramenta de fácil manuseio com recursos limitados.

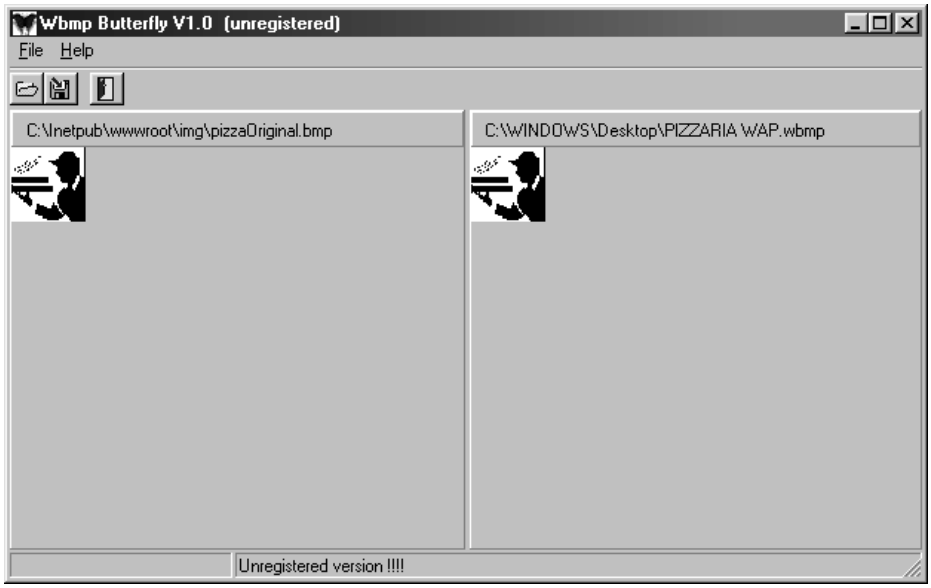

Figura 5.8 – Butterfly

b) Dissect Image

É uma ferramenta gráfica *freeware*, utilizado para criação de arquivos do formato WBMP, capacidade de processamento de imagens inteligentes. Suporta a maioria dos formatos de imagem - BMP, GIF, ICO, JPEG e WBMP.

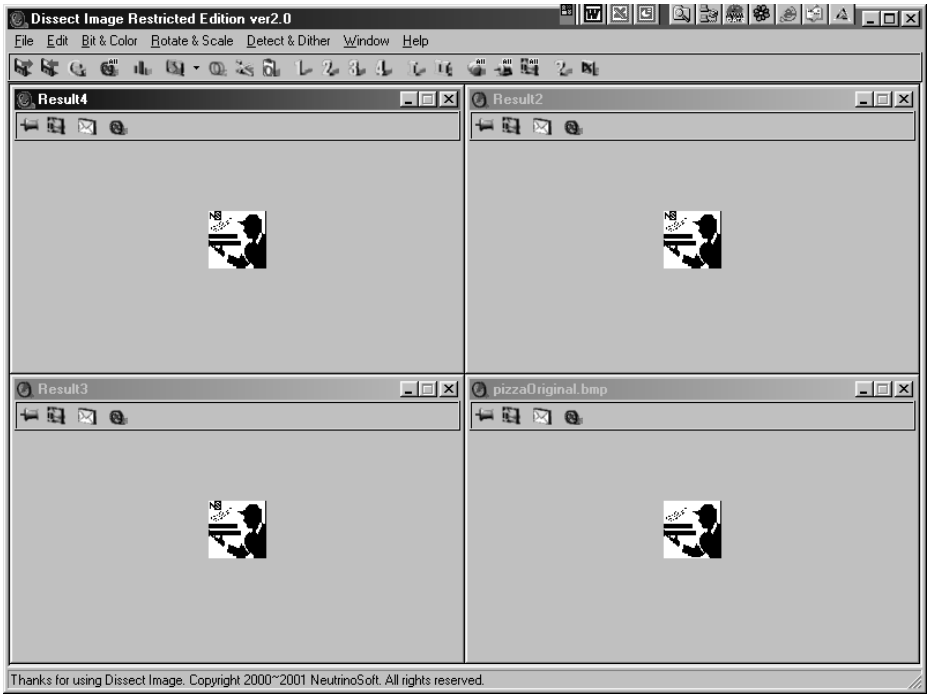

Figura 5.9 – Dissect Image

c) Generator Wbmp

Programa f*reeware*, possui recursos para conversão e criação de arquivos *wbmp* para outros formatos de arquivos. Abre imagens de vários formatos de arquivos: JPG, GIF, BMP e WBMP. Conversor de cores para: escala de cinza ou preto e branco. Altera tamanho das imagens. Possui recurso para edição das imagens ponto-a-ponto. No próprio programa o desenvolvedor pode escolher entre oito idiomas diferentes para trabalhar com a ferramenta.

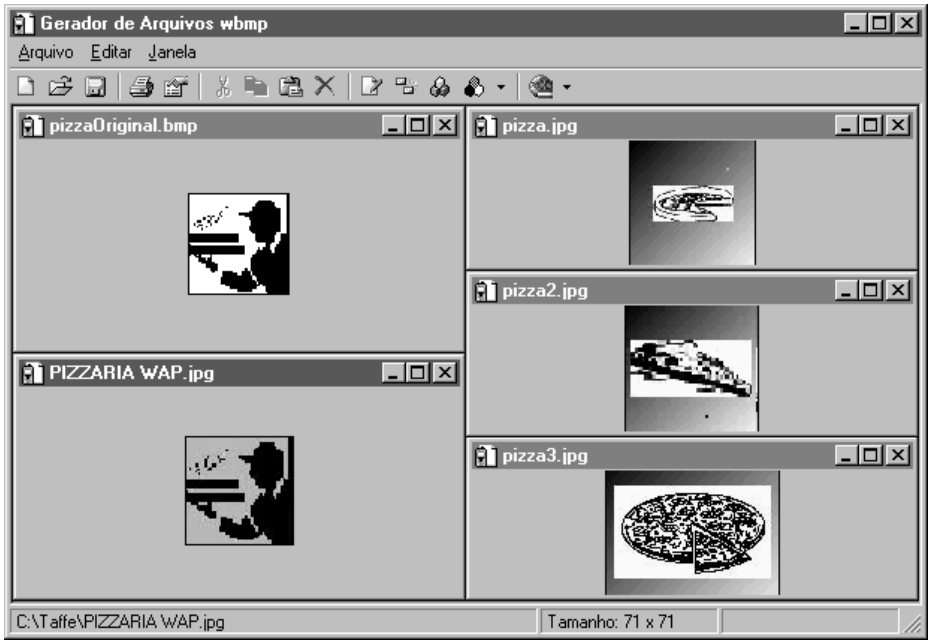

Figura 5.10 – Generator Wbmp

d) Pic2Wbmp

É uma ferramenta *freeware,* utilizada para conversão de imagens para o formato WBMP, de fácil utilização. É necessário que o Java Runtime Environment 1.1 (ou superior) esteja instalado na máquina para sua utilização.

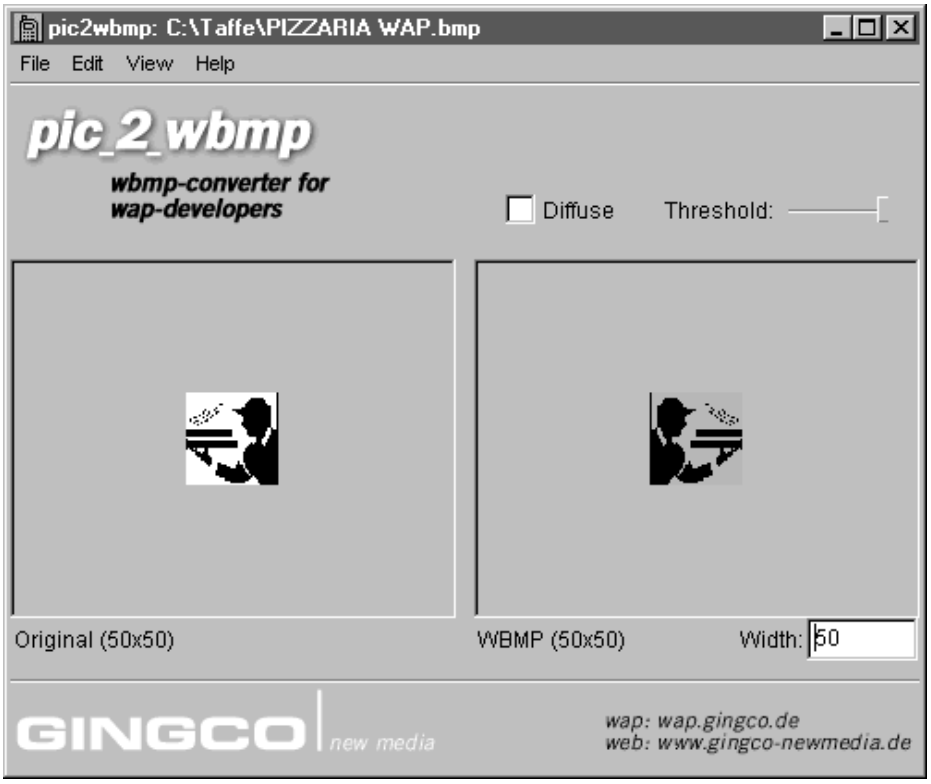

Figura 5.11 – Pic2Wbmp

e) Wap Draw

Programa para criação de imagens no formato *WBMP*, as imagens podem ser de até 96 x 80 pixels, possui um visualizador para a imagem que está em desenvolvimento. É um programa *shareware*, que roda nas plataformas: Windows 95, 98, 2000 e NT.

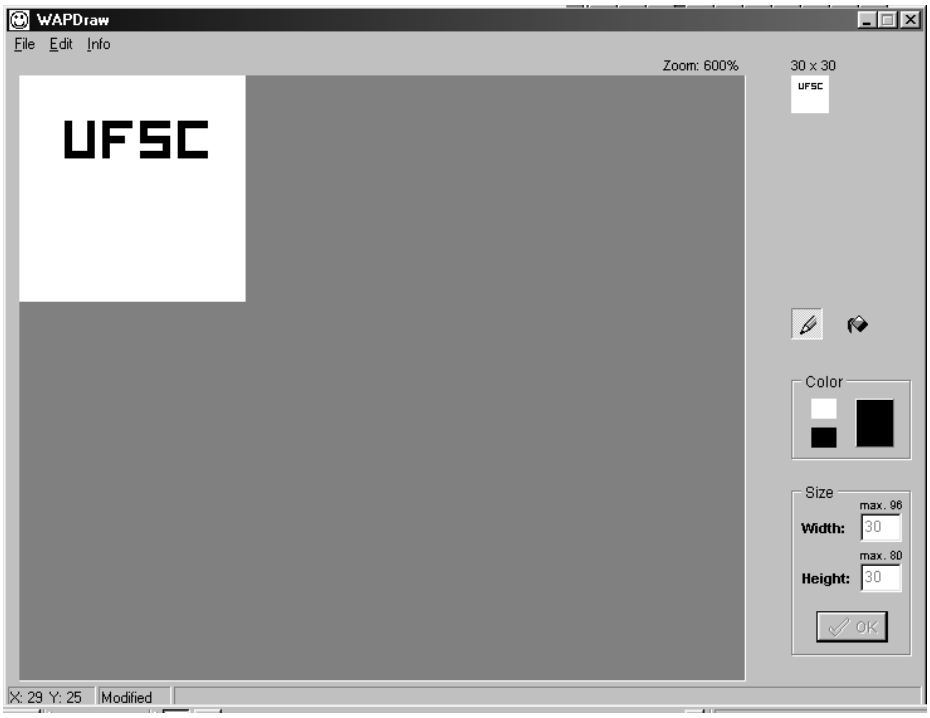

Figura 5.12 – Wap Draw

f) Wap Pictus

É um editor e gerenciador de imagens *freeware*, possui recurso de conversão *on-line*. Converte imagens no formato: *BMP, GIF, JPEG* e outros para o formato *WBMP*. Possui recurso para a escolha de um aparelho celular ou configuração das dimensões de uma tela qualquer.

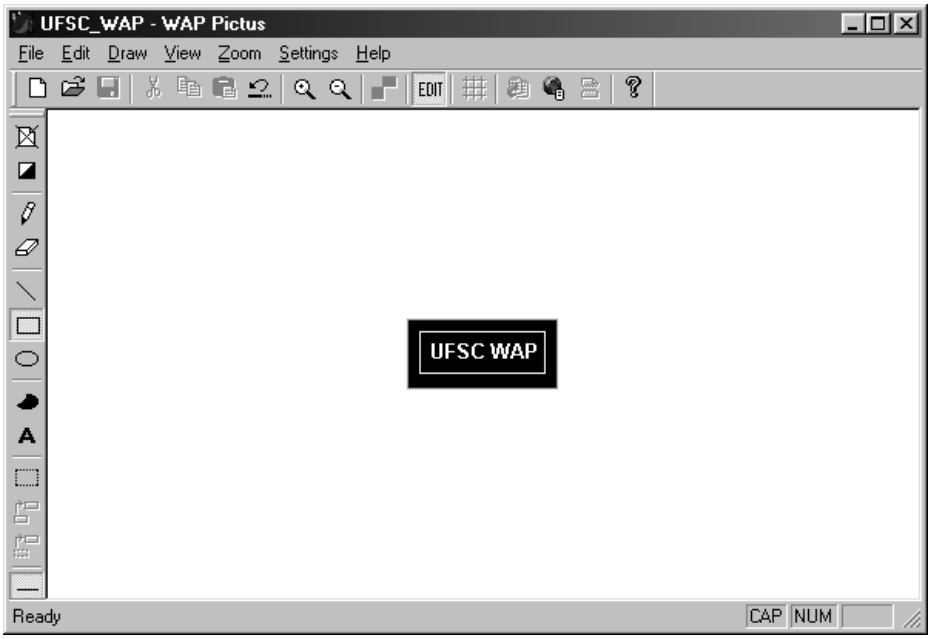

Figura 5.13 – Wap Pictus

5.3 Micronavegadores

Para se fazer à navegação na *Web* tradicional utilizamos navegadores como: Internet Explorer e Netscape. Já nos dispositivos móveis a navegação é feita de forma semelhante, utilizando micronavegadores (*microbrowsers*). São desenvolvidos principalmente pelas próprias empresas de aparelhos celulares (Nokia, Motorola, Ericsson e outras) (ARAUJO, 2001).

Segundo DEMÉTRIO (2000), os micronavegadores são responsáveis pela interpretação das linguagens *WML* e *WML Script* e sua exibição na tela dos dispositivos. Alguns micronavegadores possuem recursos de pesquisa e armazenamento das páginas mais acessadas (FAVORITOS).

Ainda segundo ARAUJO (2001), o acesso a informações também são feitos de forma diferenciada da internet tradicional. Onde dispositivos não acessam diretamente as páginas WML, pois, recebem os dados compilados por um *Gateway Wap*.

5.4 Emuladores

Emuladores são ferramentas que permitem a execução de códigos *wml* e a demonstração dos resultados, simulando diversas telas de aparelhos celulares. Ganhando com isso uma maior flexibilidade, pois, seria impossível efetuar os testes diretamente em todos dispositivos móveis existentes (FACUNTE, 2000).

Segundo RUSEYEV (2001), os emuladores são programas que têm a capacidade de simulação de diversas telas de celulares em seu computador. Mas não esquecendo da importância de ampliação dos testes a outros emuladores.

Os principais motivos para utilizar emuladores (FORTA et al, 2000, HEIJDEN e TAYLOR, 2000):

- A utilização do emulador é a maneira mais rápida de testar um código WML novo ou revisado;
- A maioria dos emuladores também oferece opções úteis de depuração (como a capacidade de inspecionar variáveis ou fonte recuperada);
- Os emuladores permitem testar as telas em vários dispositivos – mais dispositivos do que você poderia realmente ter;
- Alguns emuladores oferecem ambientes completos de desenvolvimento, que você pode usar para simplificar e aprimorar o processo de desenvolvimento.

Entre os principais emuladores destacam-se:

- Ericsson WAPIDE SDK;
- Mobile Application Development Kit (ADK) – Motorola;
- Nokia WAP Toolkit;
- UP. Simulator - Phone.com.

Capítulo 6

PROTÓTIPO – Pizzaria WAP

O protótipo está baseado numa pizzaria virtual com o nome: Pizzaria WAP. Para utilização do serviço “Pizza-Wap”, foi considerada a situação onde o cliente efetuará primeiramente um cadastro na própria pizzaria (física). Por diversos fatores como: dificuldade no preenchimento do cadastro utilizando o teclado do dispositivo móvel, garantia por parte da pizzaria que o cadastro não seja um trote e explicação detalhada sobre os procedimentos adotados para utilização do serviço.

A linguagem dinâmica utilizada foi o ASP, principalmente pela facilidade na construção de aplicativos dinâmicos, simplicidade no aprendizado, fácil manutenção e flexibilidade (podendo ser utilizadas diferentes linguagens *script* como: JScript, VBScript, e PerlScript . O ASP permite uma ótima interação entre servidor web e o banco de dados e outros sistemas como: e-mail e sistemas de arquivos. Para utilização do ASP em páginas WAP, é feita a conversão dos arquivos .wml para .asp e a especificação o tipo MIME no início do arquivo (conforme visto na seção 4.4).

O banco de dados utilizado foi o Access 2000, o protótipo por se tratar de um aplicativo de pequeno porte, o Access atendeu a necessidade do aplicativo. Poderia também ser utilizados outros bancos de dados conhecidos como: SQL Server, Oracle e MYSQL. Foram utilizadas neste protótipo cinco tabelas: borda, cliente, pedido, pizza e refrigerante. A interação com o banco de dados Access 2000 foi efetuada via ODBC.

A ferramenta de edição de imagem utilizada que apresentou maior funcionalidade foi o Generator Wbmp (seção 5.2c). Não somente pelo fato de possuir recurso de conversão em vários formatos de imagens existentes, pois a maioria dos softwares apresentados na seção 5.2 possui esse recurso. Mas principalmente pelo recurso de edição ponto-a-ponto da imagem, onde foi trabalhada minuciosamente a imagem para melhor se adaptar a tela do dispositivo. Referente a utilização de imagens estáticas e

animadas foram restritas devido às dificuldades como: velocidade de conexão, capacidade de processamento dos dispositivos, telas de tamanhos reduzidos.

O editor WML escolhido foi o WapTor, apresentado na seção 5.1f, por se tratar de uma ferramenta *freeware,* de fácil manuseio, multiplataforma, possui simulador, suporte *on-line* para todas a tags existentes e que mais se adaptou a necessidade do protótipo. Foi utilizada também a ferramenta Rasquares WapPro (seção 5.1e), por tempo limitado, sendo um aplicativo completo para edição wml, rodando em várias plataformas, possui flexibilidade para edição das principais linguagens dinâmicas do mercado (ASP, Cold Fusion, JSP, Perl, PHP, entre outros) e principalmente, interatividade na construção do código WML, porém, utilizados em projetos de grande porte.

Para simulação foram utilizados os principais emuladores: Nokia, Motorola, Ericsson e Phone.com, analisados na seção 5.4. Sendo o emulador da Phone.com o que apresentou maior confiabilidade, bom desempenho e possuindo várias opções para emulação de aparelhos celulares. Porém, é recomendável que após a conclusão de um site WAP, o mesmo deve passar por uma série de testes em emuladores, para evitar problemas como: limitação de tela, falta de suporte a imagens, diferença de *browsers,*entre outros.

6.1 Tela Inicial

A tela inicial é composta basicamente por um título, logotipo e um corpo de textos, conforme mostra figura 6.1. A imagem trabalhada no logotipo anteriormente encontrava-se no formato JPG, sendo esta convertida para o formato WBMP (*Wireless Bitmap*) padrão de imagem para dispositivos móveis. Utilizou-se em seu código de programação o elemento <timer>, onde após quatro segundos passará automaticamente ao *card* (cartão) seguinte.

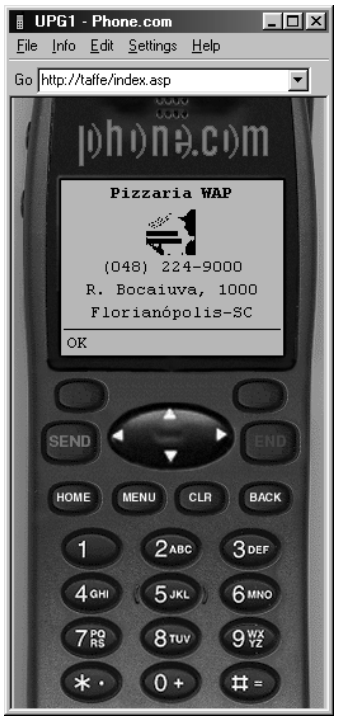

Figura 6.1 – Tela de abertura

6.2 Opções de Menu

Após a breve passagem pela tela inicial, o cartão chamado é o de Opções de Pizza (figura 6.2), onde o usuário fará sua escolha, digitando o número da pizza desejada ou através do botão de seleção do dispositivo. Os sabores existentes são meramente ilustrativos, sendo que podem ser acrescentados com facilidade no banco de dados do sistema, pelo próprio administrador do sistema.

O próximo passo será a escolha do tamanho da pizza (figura 6.3), nos três tamanhos de pizzas existentes: pequena, média e grande.

Figura 6.2 – Escolha da Pizza

Figura 6.3 – Escolha do Tamanho

Na tela seguinte (figura 6.4), será feita a escolha da borda da pizza e logo após (figura 6.5) a escolha do refrigerante. Sendo possível nestas etapas escolher uma opção nula (vazia), caso o cliente não queira uma pizza com borda e/ou nenhum refrigerante.

Figura 6.4 – Escolha da Borda

Figura 6.5 – Escolha do Refrigerante

6.3 Lista de Compras

Ao término dos pedidos, será exibida na tela a Lista de Compras, conforme mostra a figura 6.6. Possuindo duas opções ao cliente: confirmar o pedido ou fazer uma alteração em sua lista. Caso seja escolhida a opção ALTERAR, será reiniciado o processo de seleção, caso contrário, se o pedido estiver correto deverá ser escolhida a opção CONFIRMAR, passando a tela de identificação do cliente.

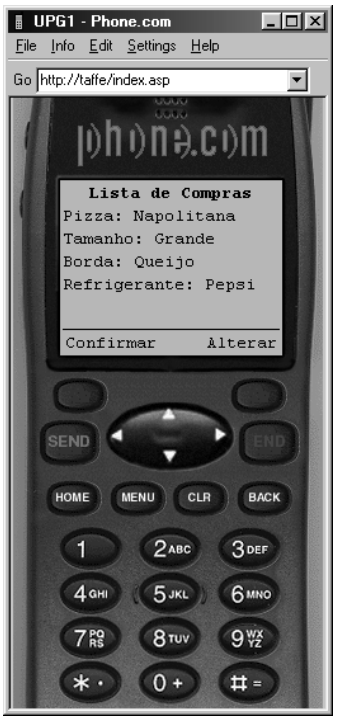

Figura 6.6 – Lista de Compras

6.4 Identificação do Cliente

Na tela de identificação, o cliente informará seu código (figura 6.7) e sua senha (figura 6.8), que são previamente definidos no cadastramento do cliente. Sendo enviadas essas informações ao sistema, onde será feita uma verificação no banco de dados da pizzaria. Retornado uma resposta a esta solicitação.

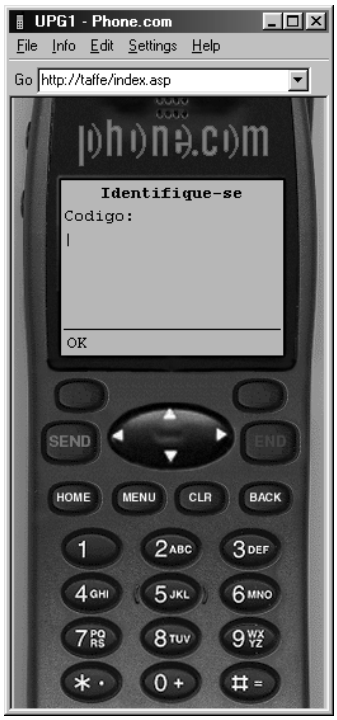

Figura 6.7 – Código do Cliente

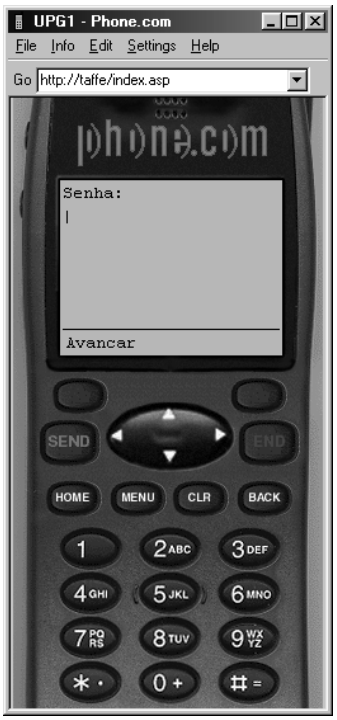

Figura 6.8 – Senha do Cliente

6.5 Erro de Identificação

Ao digitar o código do cliente errado e/ou senha, o programa informará ao usuário a seguinte tela (figura 6.9). Contendo duas opções: VOLTAR ou DISCAR. Caso seja escolhida a primeira opção, poderão ser digitados novamente seu código e senha (figura 6.7 e 6.8). Caso persistindo o erro, o cliente escolher a opção DISCAR, será efetuada uma discagem automática a pizzaria, para sanar suas dúvidas (figura 6.10).

Figura 6.9 – Problemas de Validação

Figura 6.10 – Discagem Automática

6.6 Opções de Entrega

Para agilizar o processo de entrega, o cliente no preenchimento do cadastro na pizzaria (física), informará somente um endereço para entrega do pedido. Sendo possível futuramente ser dada a opção de outros endereços.

6.6.1 Normal

Depois de confirmado o código e a senha corretamente, o cliente poderá fazer a seguinte opção de entrega: Normal ou Programada (figura 6.11). Sendo escolhida a entrega Normal, o cliente opta por receber seu pedido o mais breve possível e aparecerá na tela a mensagem de confirmação e agradecimento, conforme ilustra a figura 6.12.

Figura 6.11 – Opções de Entrega

Figura 6.12 – Tela de Confirmação

6.6.2 Programada

Caso o cliente optar pela compra Programada (figura 6.11), deverá preencher os seguintes dados: dia, mês e horário de entrega (figura 6.13). Aparecendo na seqüência a tela com a mensagem de confirmação e agradecimento (figura 6.12).

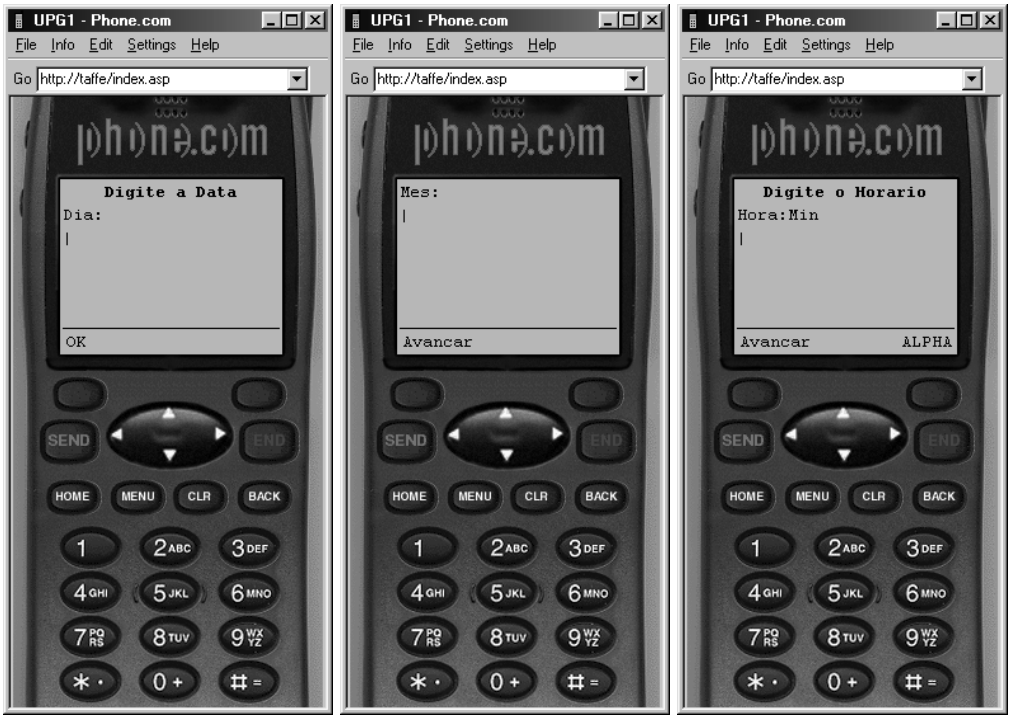

Figura 6.13 – Compra Programada

Capítulo 7

CONCLUSÃO

Acreditando no potencial crescimento do *Mobile Commerce*, empresas de telefonia móvel, fabricantes de hardware e software, desenvolvedores de conteúdo, viram com bons olhos a chegada da tecnologia *wireless* - sem fio, prova disso são os altos investimentos efetuados por empresas desses segmentos. Automação das equipes de vendas, videoconferência via dispositivo sem fio, serviços de emergência (através de localização geográfica), entretenimento, monitoração de alarmes contra roubo e incêndio, compra de produtos, são alguns exemplos de serviços disponíveis para os usuários móveis. Devido à rápida disseminação dessa tecnologia, surgiram no mercado uma grande variedade de *softwares* para desenvolvimento de *sites* para dispositivos *wireless* como: editores WML, editores de imagens e emuladores.

O presente trabalho teve por objetivo analisar e implementar ferramentas para o desenvolvimento de sites de comércio eletrônico móvel. No entanto, é preciso ressaltar alguns pontos importantes que surgiram ao longo do trabalho.

Para que houvesse uma harmonia entre os diversos dispositivos e *softwares* existentes no mercado sem fio, foi adotado como padrão o protocolo WAP – Wireless Application Protocol, este gerenciado pela WAP FÓRUM, consórcio das principais empresas ligadas a tecnologia *wireless*, que atenderia perfeitamente fatores como: interface com o usuário, limitações de processamento e *hardware*.

Para utilização dos aplicativos móveis, os dispositivos deverão possuir um micronavegador e a operadora de telefonia móvel habilitada a prover acesso ao WAP. Onde o usuário poderá solicitar informações num *site* WAP, através de um servidor Gateway ou acessar conteúdo HTTP, mas neste caso a solicitação é enviada pelo dispositivo WAP ao Gateway WAP, que transforma o conteúdo WML (linguagem de marcação sem fio) em HTTP, e posteriormente enviada ao servidor HTTP. Sendo processada a informação solicitada, esta é enviada ao Gateway WAP, que transformará o conteúdo em linguagem WML e enviando ao dispositivo móvel. Mas é importante

ressaltar que para os conteúdos possam ser visualizados corretamente nos dispositivos móveis, são necessárias que extensões do tipo MIME, estejam configuradas no servidor.

Referente a segurança no WAP, segundo DEMÉTRIO (2000), a WTLS agrega um bom nível de segurança, sendo dividida em dois momentos: no primeiro, o dispositivo móvel trocando informações com o Gateway WAP através do protocolo WTLS. E no segundo, o gateway WAP utilizando o protocolo SSL para garantir uma comunicação segura com o servidor Web. A duração do processo de conversão entre WTLS e SSL, é breve, porém, FORTA et al (2000) adverte, que a vulnerabilidade nesse pouco espaço de tempo pode ser suficiente para uma invasão. Neste caso, alguns cuidados são necessários como: utilização de um *firewall*, restrição do acesso físico ao equipamento, exclusão dos arquivos decodificados da memória volátil rapidamente e outros cuidados básicos de segurança, porém, a criptografia de ponta-a-ponta, seria fundamental para garantia da segurança nas transações *wireless*.

A linguagem de WML - *Wireless Markup Language* (Linguagem de Marcação Sem Fio), é baseada em XML (Extensible Markup Language), foi projetada para atender alguns requisitos como: ambientes que possuem limitação de hardware, baixa largura de banda, recursos de entrada e saída limitados, conexões instáveis e baixa capacidade de processamento e armazenamento. Para dar maior dinamismo à linguagem WML, foi criada a WMLScript (Wireless Markup Language Script), linguagem de script cliente orientada a objetos, dando maior velocidade ao processamento devido às bibliotecas que são armazenadas no próprio navegador do dispositivo móvel.

Para suprir a falta de recurso de animação de imagens, pois, o formato wbmp não suporta animação como *gifs* animadas, a saída encontrada seria a criação de animações quadro a quadro, semelhantes aos desenhos animados.

Foram avaliadas várias ferramentas disponíveis no mercado, e posteriormente, efetuado um prévio refinamento. Um manuseio exaustivo das ferramentas e o contato com profissionais das mais diversas áreas do conhecimento e tecnologias para montar um corpo de conhecimento interessante a respeito das formas de tecnologias utilizadas na comunicação móvel, bem como avaliar na prática o seu real funcionamento através de um protótipo, deram um escopo importante para o enriquecimento do trabalho.

Na tabela 7.1, apresenta os principais editores para linguagem WML, analisadas no trabalho, bem como suas principais vantagens e desvantagens apresentadas.

Editores para linguagem	Vantagens	Desvantagens
Adobe GoLive	Oferece na versão 6, fácil integração com WAP e iMode	Shareware, tamanho do programa, não possui emulador
Allaire Home Site	Oferece suporte integrado a linguagem WML	Shareware, não possui emulador
Inetis DotWap	Freeware, permite construção de sites de forma simples e rápida.	Não possui flexibilidade para projetos de grande porte
Macromedia Dremweaver	Oferece suporte para criação e gerenciamento de cards e decks, apresenta simulação de vários aparelhos	Shareware, não possui suporte WML nativo, não possui emulador
Rasquares WapPro	Oferece recurso para edição de código de WML dinâmico em várias linguagens, multiplataforma, interativo, possui conversor automático de imagens para o formato wbmp, suporte para grandes projetos	Shareware
WapTop EasyPad WapTor	Freeware, multiplataforma, fácil manuseio, possui emulador, suporte on-line para todas tags existentes	Não possui flexibilidade para projetos de grande porte

Tabela 7.1 – Editores para linguagem WML

Na tabela 7.1, apresenta os principais editores para linguagem WML, analisadas no trabalho, bem como suas principais vantagens e desvantagens apresentadas.

Editores de Imagens	Vantagens	Desvantagens
Butterfly	Fácil manuseio, aceita conversão de imagem de vários programas	Shareware, possui limitação de recursos
Dissect Image	Freeware, programa de fácil manuseio	Possui limitação de recursos
Generator Wbmp	Freeware, possui conversor de cores, recurso de edição de imagem ponto-a-ponto, várias opções de idiomas	-
Pic2Wbmp	Freeware, facilidade na utilização	É necessário possuir o Java Runtime Environment instalado na máquina
Wap Draw	Possui visualizador de imagem, multiplataforma	Shareware
Wap Pictus	Freeware, possui recurso de visualização de imagem em alguns modelos de celulares	Possui limitação de recursos

Tabela 7.2 – Editores de Imagens

Considerando a alta taxa de mortalidade de novas tecnologias e os custos de desenvolvimento das mesmas, é necessário estar em sintonia com as novas exigências e tendências atuais e potenciais de mercado. No sentido de visualizar as forças e fraquezas associadas ao tipo de tecnologia que melhor se adaptaria a estes requisitos e com base nos conhecimentos adquiridos a partir das análises de diversas ferramentas para implementação, desenvolveu-se um protótipo, atendendo alguns requisitos como: custo-benefício aliado à qualidade e garantia de funcionamento dos serviços prestados.

A linguagem dinâmica que melhor se adaptou a esses requisitos foi o ASP, devido à facilidade na construção de aplicativos dinâmicos, simplicidade no aprendizado, fácil manutenção e flexibilidade. Para uma melhor interação e pelo fato do protótipo não ser um sistema que exigisse um banco de dados mais robusto, utilizou-se o Access 2000 da Microsoft, sendo a interação com o banco de dados Access efetuada via ODBC. A ferramenta de edição de imagem utilizada foi o Generator Wbmp, por ser um programa completo de recursos de imagens, como por exemplo, a edição ponto-a-ponto, onde se trabalha com maiores detalhes a imagem para melhor se adaptar a tela do dispositivo. O editor WML WapTor, aliou custo-benefício com qualidade, sendo de fácil manuseio, possuindo vários recursos como: simulador, multiplataforma e suporte *on-line* para todas a tags. Após a conclusão de um *site* WAP, é aconselhável executar uma série de

testes em emuladores (simuladores), para garantir usabilidade para uma gama de usuários cada vez maior. Para simulação foram utilizados os principais emuladores: Nokia, Motorola, Ericsson e Phone.com. Sendo o emulador da Phone.com o que apresentou maior confiabilidade e bom desempenho.

Para que o Comércio Eletrônico Móvel se consolide no mercado mundial, cabem as empresas estarem fomentando meios de gerar experimentação por parte dos usuários, através de pesquisas relacionadas à utilização e principais problemas associados a seu uso, bem como buscar informações a respeito das vantagens e desvantagens de tal tecnologia e como traduzi-las em produtos e serviços que sejam percebidos como dotados de credibilidade tanto pelos usuários comuns, quanto para os usuários corporativos, sendo assim um grande diferencial competitivo.

7.1 Trabalhos futuros

Apesar do protótipo ser implementado numa empresa de pequeno porte ou serviço para o usuário tradicional, poderia também ser implementado numa grande corporação, que é o segmento que deve absorver este mercado, mas devido à resistência apresentada e incertezas apresentadas pelo WAP, fica difícil a abertura das empresas em expor seus sistemas para testes. Sendo que ao término do trabalho possa então comprovar as reais utilidades com uma base concreta de conhecimentos sobre a viabilidade dos serviços. Apesar da escassa literatura disponível, foi possível montar um corpo de conhecimento interessante a respeito das formas de tecnologias para a comunicação móvel, bem como avaliar na prática o seu real funcionamento através de um protótipo, porém, seria de suma importância que fossem feitos estudos que apontem as viabilidades de criação de meios para garantir uma total segurança nas transações ponto-a-ponto, ferramentas para conversão de conteúdo e serviço, gerando uma melhor interação entre hardware e software. Bem como comparativos entre a tecnologia atual e a japonesa que alcançou um patamar considerável às tecnologias existentes no mercado e sua aparente vantagem quando comparada as aplicabilidades de serviços e protocolos.

ANEXOS

Anexo 1 – Arquivo INDEX

```
<% Response.ContentType="text/vnd.wap.wml" %>
<?xml version="1.0"?>
<!DOCTYPE wml PUBLIC "-//WAPFORUM//DTD WML 1.1//EN"
"http://www.wapforum.org/DTD/wml_1.1.xml">

<%
Dim conexao, sql
Set conexao = Server.CreateObject("ADODB.Connection")
ODBC = "pizzariaWap"
conexao.Open ODBC,"Administrador","123456"
%>

<wml>

<card id="principal" ontimer="#tipoPizza">
  <timer value="40"/>
  <p align="center"><em>Pizzaria WAP</em></p>
  <p align="center">
    <img alt="Pizza" src="pizza.wbmp" align="top"/>
    <br/> (048) 9107-1313 <br/>
     R. Bocai&#250;va, 1313 <br/>
     Florian&#243;polis-SC <br/>
  </p>
</card>

<card id="tipoPizza">
  <do type="accept" label="Avancar">
          <go href="#tamanhoPizza"/>
  </do>
  <p align="center"><em>Escolha a Pizza</em></p>
```

```
  <p>
        <select name="pizza">

        <%
        Set resultado = conexao.Execute("SELECT * FROM pizza")
        While not resultado.EOF
        %>

                        <option
value="<%=resultado("nome")%>"><%=resultado("nome")%></option>

                <%
                resultado.MoveNext
        WEnd
        %>

        </select>
  </p>
</card>

<card id="tamanhoPizza">
  <do type="accept" label="Avancar">
        <go href="#bordaPizza"/>
  </do>
  <p align="center"><em>Tamanho da Pizza</em></p>
  <p>
        <select name="tamanho">
                <option value="Pequena">Pequena</option>
                <option value="Media">M&#233;dia</option>
                <option value="Grande">Grande</option>
        </select>
  </p>
</card>

<card id="bordaPizza">
```

```
<p align="center"><em>Borda da Pizza</em></p>
<do type="accept" label="Avancar">
        <go href="refrigerante.asp" method="post">
                <postfield name="pizza" value="$pizza"/>
                <postfield name="tamanho" value="$tamanho"/>
                <postfield name="borda" value="$borda"/>
        </go>
</do>
<p>
        <select name="borda">

        <%
        Set resultado = conexao.Execute("SELECT nome FROM borda")
        While not resultado.EOF
        %>

                        <option
value="<%=resultado("nome")%>"><%=resultado("nome")%></option>

                <%
                resultado.MoveNext
        WEnd
        %>

        </select>
</p>
</card>

</wml>
```

Anexo 2– Arquivo CONFIRMAÇÃO

```
<% Response.ContentType="text/vnd.wap.wml" %>
<?xml version="1.0"?>
<!DOCTYPE wml PUBLIC "-//WAPFORUM//DTD WML 1.1//EN"
"http://www.wapforum.org/DTD/wml_1.1.xml">

<%
Dim conexao, sql
Set conexao = Server.CreateObject("ADODB.Connection")
ODBC = "pizzariaWap"
conexao.Open ODBC,"Administrador","123456"

Session("refri") = request.Form("refrigerante")
%>

<wml>

<card id="compras">
  <p align="center"><em>Lista de Compras</em></p>
  <do type="accept" label="Confirmar">
        <go href="#login"/>

  </do>
  <do type="accept" label="Alterar">
        <go href="index.asp#tipoPizza"/>
  </do>
  <p>
        Pizza: <%=session("pizza")%><br/>
        Tamanho: <%=session("tamanho")%><br/>
        Borda: <%=session("borda")%><br/>
        Refrigerante: <%=session("refri")%>
  </p>
</card>
```

```
<card id="login">
  <p align="center"><em>Identifique-se</em></p>
  <do type="accept" label="Avancar">
        <go href="verificaLogin.asp" method="post">
              <postfield name="cliente" value="$cliente"/>
              <postfield name="senha" value="$senha"/>
        </go>
  </do>
  <p>
        Codigo:
        <input name="cliente" type="text" format="6N" emptyok="false"/> <br/>
        Senha:
        <input name="senha" type="password" format="6N" emptyok="false"/>
  </p>
</card>


      </wml>
```

Anexo 3 - Arquivo ENVIAR

```
<% Response.ContentType="text/vnd.wap.wml" %>
<?xml version="1.0"?>
<!DOCTYPE wml PUBLIC "-//WAPFORUM//DTD WML 1.1//EN"
"http://www.wapforum.org/DTD/wml_1.1.xml">

<wml>

<%
Dim conexao, sql, dataPedido
Set conexao = Server.CreateObject("ADODB.Connection")
ODBC = "pizzariaWap"
conexao.Open ODBC,"Administrador","123456"
```

```
sql = "SELECT nome FROM cliente WHERE codigo = "& request.Form("cliente") &" AND
senha = "& request.Form("senha") &""
Set validacao = conexao.Execute(sql)
If validacao.EOF Then
%>

<card id="acessoNegado" ontimer="index.asp#login">
 <timer value="10"/>
 <p align="center">
 Nao confere.
 </p>
</card>

<%
Else
        dataPedido = day(now) & "/" & month(now) & "/" & year(now)
        conexao.Execute("INSERT INTO pedido (codigo_cliente, pizza, refri, dataPedido)
VALUES ("& request.Form("cliente") &",'"& request.Form("pizza") &"','"&
request.Form("refri") &"', '"& dataPedido &"')")
%>

<card id="MainCard" title="Conferindo">
 <p align="center">
 <em>Confirmado</em> <br/>
Estaremos entregando o seu pedido em 30 minutos. <br/>
Obrigado!
 </p>
</card>

<% End If %>

</wml>
```

Anexo 4 - Arquivo ENCERRAMENTO

```
<% Response.ContentType="text/vnd.wap.wml" %>
<?xml version="1.0"?>
<!DOCTYPE wml PUBLIC "-//WAPFORUM//DTD WML 1.1//EN"
"http://www.wapforum.org/DTD/wml_1.1.xml">

<wml>

<%
Dim conexao, dataPedido, entrega
Set conexao = Server.CreateObject("ADODB.Connection")
ODBC = "pizzariaWap"
conexao.Open ODBC,"Administrador","123456"

If ( Session("cliente") <> "" ) Then
        If ( request.QueryString("entrega") = "normal" ) Then
                entrega = "normal"
                dataEntrega = day(now) & "/" & month(now)
                horario = ""
        Else
                entrega = "programada"
                dataEntrega = request.Form("dia") & request.Form("mes")
                horario = request.Form("horario")
        End If

        dataPedido = day(now) & "/" & month(now) & "/" & year(now)
        'conexao.Execute("INSERT INTO pedido (codigo_cliente, pizza, tamanho, borda,
refrigerante, tipoEntrega, dataPedido, dataEntrega, horario) VALUES ("& Session("cliente")
&",'"& Session("pizza") &"','"& Session("tamanho") &"','"& Session("borda") &"','"&
Session("refrigerante") &"','"& entrega &"','"& dataPedido &"','"& dataEntrega &"','"& horario
&"')")
        'sql = "INSERT INTO pedido (codigo_cliente, pizza, tamanho, borda, refrigerante,
tipoEntrega, dataPedido, dataEntrega, horario) VALUES ("& Session("cliente") &",'"&
```

```
Session("pizza") &"','"& Session("tamanho") &"','"& Session("borda") &"','"& Session("refri")
&"','"& entrega &"','"& dataPedido &"','"& dataEntrega &"','"& horario &"')"
        conexao.Execute("INSERT INTO pedido (codigo_cliente, pizza, tamanho, borda,
refrigerante, tipoEntrega, dataPedido, dataEntrega, horario) VALUES ("& Session("cliente")
&",'"& Session("pizza") &"','"& Session("tamanho") &"','"& Session("borda") &"','"&
Session("refri") &"','"& entrega &"','"& dataPedido &"','"& dataEntrega &"','"& horario &"')")
%>

<card>
 <p align="center">
<big><i>Confirmado</i></big><br/>
<u>Estaremos entregando o seu pedido conforme solicitado.</u><br/><br/>
Obrigado!
 </p>
<p><%=sql%></p>
</card>

<% Else %>

<card>
  <do type="accept" label="Voltar">
        <go href="confirmacao.asp#login"/>
  </do>
  <do type="accept" label="Discar">
    <go href="wtai://wp/mc;04591071313"/>
  </do>
 <p align="center">
 Seu codigo e/ou senha nao conferem.
 </p>
</card>

<% End If %>

</wml>
```

Anexo 5 - Arquivo REFRIGERANTE

```
<% Response.ContentType="text/vnd.wap.wml" %>
<?xml version="1.0"?>
<!DOCTYPE wml PUBLIC "-//WAPFORUM//DTD WML 1.1//EN"
"http://www.wapforum.org/DTD/wml_1.1.xml">

<%
Dim conexao, sql
Set conexao = Server.CreateObject("ADODB.Connection")
ODBC = "pizzariaWap"
conexao.Open ODBC,"Administrador","123456"

Session("pizza") = request.Form("pizza")
Session("tamanho") = request.Form("tamanho")
Session("borda") = request.Form("borda")
%>

<wml>

<card id="tipoRefrigerante">
  <do type="accept" label="Avancar">
        <go href="confirmacao.asp" method="post">
              <postfield name="refrigerante" value="$refrigerante"/>
        </go>
  </do>
  <p align="center"><em>Escolha o Refrigerante</em></p>
  <p>
        <select name="refrigerante">

        <%
        Set resultado = conexao.Execute("SELECT * FROM refrigerante")
        While not resultado.EOF
        %>
```

```
                    <option
value="<%=resultado("nome")%>"><%=resultado("nome")%></option>

                <%
                resultado.MoveNext
        WEnd
        %>

        </select>
  </p>
</card>

</wml>
```

Anexo 6 – Verifica Login

```
<% Response.ContentType="text/vnd.wap.wml" %>
<?xml version="1.0"?>
<!DOCTYPE wml PUBLIC "-//WAPFORUM//DTD WML 1.1//EN"
"http://www.wapforum.org/DTD/wml_1.1.xml">

<wml>

<%
Dim conexao, cliente, senha
If ( request.Form("cliente") <> "" ) Or ( request.Form("senha") <> "" ) Then
        cliente = request.Form("cliente")
        senha = request.Form("senha")
Else
        cliente = 0
        senha = 0
End If
```

```
Set conexao = Server.CreateObject("ADODB.Connection")
ODBC = "pizzariaWap"
conexao.Open ODBC,"Administrador","123456"

Set validacao = conexao.Execute("SELECT nome FROM cliente WHERE codigo = "& cliente
&" AND senha = "& senha &"")
If validacao.EOF Then
%>

<card id="acessoNegado">
  <do type="accept" label="Voltar">
          <go href="confirmacao.asp#login"/>
  </do>
  <do type="accept" label="Discar">
    <go href="wtai://wp/mc;045 91071313"/>
  </do>
 <p align="center">
 <b>ATEN&#199;&#195;O !!!</b><br/><br/>
<img alt="" localsrc="multiply" src=""></img><br/><br/>
<b>C&#243;digo e/ou senha<br/>
 n&#227;o conferem.</b>
 </p>
</card>
<%
Else
          Session("cliente") = cliente
%>
<card id="tipoEntrega">
  <p align="center"><em>Opcoes de entrega</em></p>
  <p>
          <select name="entrega">
                    <option onpick="encerramento.asp?entrega=normal"
value="Normal">Normal</option>
                    <option onpick="#data" value="Programada">Programada</option>
          </select>
```

```
  </p>
</card>

<card id="data">
  <do type="accept" label="Avancar">
        <go href="#horario"/>
  </do>
  <p align="center"><em>Digite a Data</em></p>
  <p>
        Dia:<input name="dia" type="text" format="2N" emptyok="false"/> <br/>
        Mes:<input name="mes" type="text" format="2N" emptyok="false"/> <br/>
  </p>
</card>

<card id="horario">
  <do type="accept" label="Avancar">
        <go href="encerramento.asp" method="post">
              <postfield name="dia" value="$dia"/>
              <postfield name="mes" value="$mes"/>
              <postfield name="horario" value="$horario"/>
        </go>
  </do>
  <p align="center"><em>Digite o Horario</em></p>
  <p>
        Hora:Min<input name="horario" type="text" emptyok="false"/>
  </p>
</card>

<% End If %>

</wml>
```

GLOSSÁRIO

EIR – Registro de Identidade de Equipamentos

HLR – Registro de Localização na área original

VLR – Registro de Localização de Visitante

1G – Primeira Geração

2G – Segunda Geração

2,5G – Geração intermediária

3G – Terceira Geração

4G – Quarta Geração

AMPS – Sistema Avançado de Telefonia

CDMA – Acesso Múltiplo por Divisão Código

CDMA2000 – Grupo de tecnologias que atualizam as redes cdmaOne

EDGE - Taxa de Transmissão Aprimorado para Evolução Global

GPRS – Serviço Padrão para Transmissão de Pacotes Via Rádio

GSM – Sistema Global para Comunicação Móvel

HSCSD – Dados Comutados por Circuito de Alta Velocidade

NMT – Padrão de Telefone Móvel Nórdico

PDC – Celular Digital Pessoal

TACS – Sistema de Comunicação Acesso Total

TDMA – Acesso Múltiplo por Divisão de Tempo

UMTS – Sistema Universal de Telecomunicações Móveis

WATM – Modo de Transmissão Assíncrono Sem Fio

W-CDMA – CDMA de banda larga

WML – Linguagem de Marcação Sem Fio

WAP – Protocolo de Aplicações Sem Fio

WAE – Ambiente de Aplicação Sem Fio

WTA – Aplicação de Telefonia Sem Fio

WSP – Protocolo de Sessão Sem Fio

WTP – Protocolo Transacional Sem Fio

WTLS – Camada de Transporte Segura Sem Fio

WDP – Protocolo de Datagrama Sem Fio

REFERÊNCIAS BIBLIOGRÁFICAS

ALBERTIN, Alberto Luiz. Comércio Eletrônico: Modelo, Aspectos e Contribuições de sua Aplicação. São Paulo: Atlas, 1999.

ALBERTIN, Alberto Luiz. Comércio Eletrônico. São Paulo: Atlas, 2000.

ARAUJO, Jário. Desenvolvendo para WAP com WML. Rio de Janeiro: Ciência Moderna, 2001.

AREHART, Charles; HOMER, Alex; CHIDAMBARAM, Nirmal. Professional WAP. Wrox Press, 2000.

BATTISTI, Júlio. Criando Sites Dinâmicos com ASP 3.0. Rio de Janeiro: Excel Books, 2000.

CARVALHO, Alan. Desenvolvendo na Web com JavaScript. Rio de Janeiro: Book Express,2001.

CHASE, Nicholas. Aprendendo Active Server Pages 3.0. São Paulo: Makron Books. 2000.

DEMÉTRIO, Rinaldo.A tecnologia Wap – Aprenda a Criar Sites para Celulares com a Linguagem WML. São Paulo: Érica, 2000.

DENEGA, Marcos Antônio. Wap: tecnologia sem fio. São Paulo: Berkeley Brasil, 2000.

DIAS, Adilson de Souza. Desvendando o WMLScript. Rio de Janeiro: Ciência Moderna, 2000.

DIAS, Adilson de Souza. WAP – A Internet Sem Fios. Rio de Janeiro: Ciência Moderna, 2000.

DORNAN, Andy. Wireless Communication: o guia essencial de comunicação sem fio. Rio de Janeiro: Campus, 2001.

FACUNTE, EMERSON. WAP Guia de Tecnologia. Rio de Janeiro: Brasport, 2000.

FORTA, Ben; LAUVER, Keith; FONTE, Paul; et al. Wap Development with WML and WMLScript. Sams Publishing, 2000.

FROST, Martin. Aprendendo WML e WMLScript. Rio de Janeiro: Campus, 2001.

HARTE, Lawrence; KITKA, Roman; LEVINE, Richard; et al. 3G Wireless Demystified. McGraw-Hill Professional, 2001.

HEIJDEN, Marcel Van Der; TAYLOR, Marcus; Understanding Wap: Wireless Applications, Devices, and Services. Artech House, 2000.

HJELM, Johan. Designing Wireless Information Services. John Wiley & Sons, 2000.

YURI, Flávia. O 3G fala japonês. Info Exame. Novembro 2001 pg. 62/72-73.

MANN, Steve. Programming Applications with the Wireless Application Protocol. John Wiley & Sons,1999.

MEIRA JR, Wagner; MURTA, Cristina Duarte; RODOLFO, Sérgio Ferreira de. Comércio Eletrônico na WWW. São Paulo: IME-USP, 2000.

MELONI, Julie C.. Fundamentos de PHP. Rio de Janeiro : Ciência Moderna, 2000.

OLIVEIRA, Wilson José de. WAP Tecnologia e Segurança. Florianópolis: Visual Books, 2000.

RISCHPATER, Ray. Desenvolvendo WIRELESS para WEB. São Paulo: Makron Books, 2001.

RUSEYEV, Sergei. Wap Technology and Applications. Charles River Media, 2001

SETUBAL, Rogério. Desenvolvendo na WEB com W.A.P.: Wireless Aplication Protocol. Rio de Janeiro: Book Express,2000.

SHARMA, Chetan. Aplicações Comerciais da Internet Sem Fio. São Paulo: Makron Books, 2001.

TANENBAUM, Andrew S.. Computer Networks. Prentice Hall, 1996.

TERRY, Bernstein; BHIMANI, Anish B.; SCHULTZ, Eugene; et al. Segurança na Internet. Rio de Janeiro: Campus, 1997.

WATSON, Karli; FOO, Soo Mee; LEE, Wei Meng; et al. Beginning WAP: WML and WMLScript. Wrox Press, 2000.

WEISSINGER, A. Keyton. ASP : O Guia Essencial. Rio de Janeiro: Ciência Moderna, 2000.

WIRELESS APPLICATION PROTOCOL FORUM LTD. Official Wireless Application Protocol: The Complete Standard with Searchable CD-ROM. John Wiley & Sons, 1999.